Distinguishing Learning Rules with Brain Machine Interfaces

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite extensive theoretical work on biologically plausible learning rules, it has been difficult to obtain clear evidence about whether and how such rules are implemented in the brain. We consider biologically plausible supervised- and reinforcement-learning rules and ask whether changes in network activity during learning can be used to distinguish which learning rule is being used. In particular, we note that supervised learning requires a credit-assignment model estimating the mapping from neural activity to behavior and that, in a biological organism, this model will inevitably be an imperfect approximation of the ideal mapping, leading to a bias in the direction of the weight updates relative to the true gradient. Reinforcement learning, on the other hand, requires no credit-assignment model and tends to make weight updates following the true gradient direction. We derive a metric to distinguish between learning rules by observing changes in the network activity during learning, given that the mapping from brain to behavior is known by the experimenter. Because brain-machine interface experiments allow for perfect knowledge of this mapping, we focus on modeling a cursor-control BMI task using recurrent neural networks, showing that learning rules can be distinguished in simulated experiments using only observations that a neuroscience experimenter would plausibly have access to.

1 Introduction

In order to update synaptic weights effectively during learning, a biological or artificial neural network must solve the problem of credit assignment. That is, a neuron must infer whether it should increase or decrease its activity in order to decrease the error in the behavior that it is driving. Supervised learning (SL) provides one approach for solving the credit assignment problem by endowing the synaptic learning rule with a *credit assignment mapping*, i.e. an estimate of how each neuron's activity affects the behavioral readout. Indeed, gradient-based methods such as backpropagation for feedforward networks and backpropagation through time for recurrent networks do exactly this by multiplying a (possibly multi-dimensional) readout error by the downstream readout weights in order to update hidden-layer synaptic weights. However, because neurons in a biological neural network likely lack complete information about their downstream projections, ideal credit assignment mappings such as those used in gradient-based methods are biologically implausible–an issue known as the weight transport problem [1, 2, 3]. Recent work has shown that effective supervised learning in both feedforward networks and RNNs can still take place as long as there is a positive overlap between the true readout weights and the credit assignment matrix used for learning, providing one possible solution to this problem [4, 5].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Reinforcement learning (RL) is another approach for solving the credit assignment problem. Node perturbation algorithms correlate noise injections with scalar reward signals in order to update network weights [6, 7, 8]. Such algorithms avoid the weight transport problem, as the update rule does not explicitly require a credit assignment matrix mapping the vector error back to the recurrent weights. While the weight updates for algorithms in this family tend to be noisy, the policy-gradient theorem guarantees that they tend to follow the true gradient of the objective function (Appendix B.2, [6, 9]).

Brain-machine interfaces (BMI) provide an ideal paradigm for investigating questions about learning and credit assignment in real brains, as they enable the experimenter to define the mapping from neural activity onto behavior. For example, monkeys can be trained to control a cursor using a BMI in the motor cortex, and different manipulations of the BMI decoder can be learned in the course of a single day [10, 11] or over the course of multiple days [12, 13]. In the case of BMIs, the weight transport problem is particularly acute, since the readout weights mapping the neural activity onto the cursor position may be changed abruptly, and there is no plausible way for the agent to generate an instantaneous estimate of the new readout weights in order to then apply a learning rule. If an agent is to use a credit-assignment model at all in such a task, it must consist of a weight matrix that imperfectly approximates the readout mapping, i.e. a *biased* internal model of credit assignment.

The key insight in our work is that, under an SL rule, the agent will invariably perform credit assignment using an imperfect internal model of how its neural activity is mapped onto the kinematics of the BMI readout, whereas no such internal model is used under a policy gradient-like RL rule. We show that, for SL but not for RL, this imperfect internal model introduces a systematic bias in the weight updates, which in turn leads to a bias in the direction in which neural activity evolves during learning, thereby providing a means by which to distinguish these different learning scenarios. In this work, we develop a framework for modeling BMI experiments with recurrent neural networks (RNNs), using biologically plausible versions of SL and RL. We then show how, under certain experimental designs, different learning rules can lead to distinct changes in neural activity during the course of learning. We derive a statistical metric for comparing observed changes in neural activity with the changes predicted assuming knowledge of the BMI decoder, and we show how this metric can be used to detect learning bias in our simulated experiments.

Related work. While there has recently been a renaissance of research on biologically plausible learning rules [4, 8, 14, 15, 5, 3], there has been relatively little work on how such proposals might be experimentally verified in neural experiments. Lim et al. [16] inferred hyperparameters of a Hebbian learning rule based on neural activity from a visual cortical area in monkeys. Their work, however, assumed a single class of learning rules rather than trying to distinguish between multiple classes. Nayebi et al. [17] showed that convolutional neural networks trained with different optimizers and hyperparameters can in principle be distinguished by a classifier based on both weight changes and neural activations. However, the classifier was trained using ground-truth data about the identity of the learning rule, which is the very thing that we aim to infer in our model. Kepple et al. [18] proposed a curriculum-based method for discerning the objective being optimized with SL and showed via RNN modeling that different objectives lead to distinct task-acquisition timescales. While their approach analyzed evidence accumulation and decision tasks and focused on distinguishing different objectives, we focus primarily on continuous control tasks and focus on distinguishing learning rules. Feulner and Clopath [19] simulated BMI learning with RNNs to compare learning that required changes within vs. outside of the intrinsic manifold of neural activity, while Feulner et al. [20] used a similar approach to analyze how learning at upstream or recurrent synapses affected the covariance of neural activity, but neither of these studies attempted to distinguish learning rules. To our knowledge, our study is the first to propose a framework for distinguishing between learning rules using BMIs.

2 Results

The overall approach for our work is shown schematically in Fig. 1. We train RNNs to perform a BMI-inspired cursor-control task using different learning rules, with the goal of using the observed activity to infer the learning rule that was used to train the network. As in numerous BMI experiments

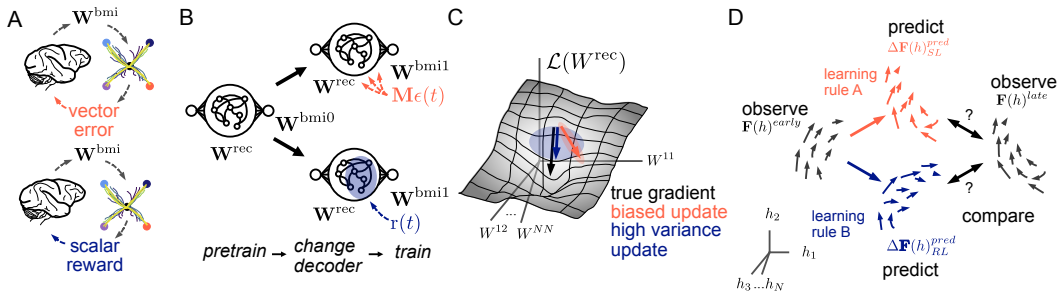

Figure 1: Approach. (A) We model a brain-machine interface experiment in which a monkey learns to move a cursor on a screen using either supervised learning (SL, top) or reinforcement learning (RL, bottom). (B) We pretrain an RNN to perform a center-out reach task with a fixed decoder $\mathbf{W}^{\text{bmi0}}$, switch the decoder to $\mathbf{W}^{\text{bmi1}}$, then train the recurrent weights $\mathbf{W}^{\text{rec}}$ again with either SL or RL. (C) SL with a biased credit assignment mapping will lead to biased recurrent weight updates, whereas RL leads to noisy but unbiased weight updates. (D) By inferring the neural activity flow fields from observed data and comparing the observed change during learning with the predicted changes assuming SL or RL, we infer which learning rule was used to train the RNN.

[10, 21, 13], we pre-train the RNN using a particular choice of decoder weights, then require the RNN to relearn the task using a new set of decoder weights. Our main result is to predict the different ways in which the RNN activity is expected to evolve through learning, then to compare these predictions with the observed changes in the activity in order to infer the learning rule. In the sections that follow, we will first introduce the SL and RL update rules that we will use for training our networks, then we will show that they lead to distinct predictions about how the RNN activity is expected to evolve with training. The remainder of the paper will present empirical results showing that the learning rules can be inferred by observing the network activity, first for a relatively simple case and then under more generalized assumptions.

While our main results can be obtained using either a feedforward or recurrent circuit model, we mostly focus on the latter case because BMI experiments typically involve control tasks driven by time-varying activity in motor cortex, for which an RNN model is more appropriate. The feedforward version of our results is provided in Appendix A.

Modeling BMI learning with RNNs. We model motor cortex as a vanilla RNN characterized by input weights $\mathbf{W}^{\text{in}}$, recurrent weights $\mathbf{W}^{\text{rec}}$, feedback weights $\mathbf{W}^{\text{fb}}$, and BMI decoder weights $\mathbf{W}^{\text{bmi}}$. Since the experimenter has complete control over the decoder in a BMI experiment, we treat $\mathbf{W}^{\text{bmi}}$ as fixed and assume that learning occurs in recurrent weights $\mathbf{W}^{\text{rec}}$ alone. The activity of the recurrent units $\mathbf{h}^t$ is given by

$$h_i^t = \Big(1 - \frac{1}{\tau}\Big) h_i^{t-1} + \frac{1}{\tau}\phi\Big(\sum_j W_{ij}^{\text{rec}} h_j^{t-1} + \sum_j W_{ij}^{\text{in}} x_j^t + \sum_j W_{ij}^{\text{fb}} y_j^t\Big) + \xi_i^t, \tag{1}$$

where $\mathbf{x}^t$ is an input signal, and $\xi_i^t \sim \mathcal{N}(0, \sigma^2)$ is isotropic noise. This activity is read out by $\mathbf{W}^{\text{bmi}}$ to obtain the cursor position $y_k^t = \sum_i W_{ki}^{\text{bmi}} h_i^t$. The error at each time is defined as $\epsilon_k^t = {y^*}_k^t - y_k^t$, where $\mathbf{y}^{*t}$ is the target output for each time step. The loss to be minimized is then $L = \frac{1}{2T}\sum_{t=1}^{T}\sum_{k=1}^{N_y}(\epsilon_k^t)^2$ where T is the trial duration and $N_y = 2$ is the number of readout dimensions.

Supervised learning. Standard supervised algorithms such as Backpropagation Through Time [22] and Real-Time Recurrent Learning [23] are widely acknowledged to be biologically implausible. The reasons for this are two-fold. First, both algorithms assume knowledge of the readout weights for purposes of credit assignment, which we referred to above as the weight transport problem. Second, updating a particular recurrent weight with one of these algorithms requires knowledge of all of the other weights in the network, which is not information that a biological neuron would likely have access to.

A recently proposed alternative, Random Feedback Local Online (RFLO) learning [5], is an approximate gradient-based algorithm with $\Delta \mathbf{W}^{\mathrm{rec}} \approx -\partial L / \partial \mathbf{W}^{\mathrm{rec}}$ that addresses these problems by (i) assuming that error information is projected back into the network with weights that imperfectly approximate the ideal credit assignment mapping, and (ii) dropping nonlocal terms from the update rule. The recurrent weight update with this learning rule is

$$\Delta W_{ij}^{\mathrm{rec}} = \eta \sum_{t=1}^{T} \left[\mathbf{M}\boldsymbol{\epsilon}^t\right]_i p_{ij}^t, \qquad p_{ij}^t = \left(1 - \frac{1}{\tau}\right) p_{ij}^{t-1} + \frac{1}{\tau}\phi'(u_i^t) h_j^{t-1}, \tag{2}$$

with $u_i^t = \sum_k W_{ik}^{\mathrm{rec}} h_k^{t-1} + \sum_k W_{ik}^{\mathrm{in}} x_k^t + \sum_k W_{ik}^{\mathrm{fb}} y_k^t$ and $p_{ij}^t \approx \partial h_i^t / \partial W_{ij}^{\mathrm{rec}}$. Further details are provided in Appendix B.1. The learning rule that we implement differs slightly from RFLO in that the readout weights, as in a BMI experiment, are fixed during training rather than learned. In RFLO, learning of the readout weights was required in order to align the readout weights with the credit assignment matrix. Here we will simply assume that $\mathbf{M}$ is partially aligned with $(\mathbf{W}^{\mathrm{bmi}})^\top$.

In (2), the credit assignment mapping M_{ik} assigns responsibility for the readout error ϵ_k^t to neuron i. While the ideal mapping would be $\mathbf{M} = (\mathbf{W}^{\mathrm{bmi}})^\top$, learning can still be effective as long as there is a positive correlation between $\mathbf{M}$ and $(\mathbf{W}^{\mathrm{bmi}})^\top$ [5] (as shown earlier in the case of feedforward networks [4]). In the case where they are not equal, as we presume they cannot be in a BMI experiment unless the monkey is very highly overtrained, the matrix $\mathbf{M}$ constitutes a *biased internal model* of credit assignment. Below, we will show that this leads to a predictable bias in the change of the RNN activity as the task is learned.

Example output trajectories from an RNN trained on the cursor-control task with the SL rule described above are shown in Fig. 2A. The results in Fig. 2B show that learning the task is possible whenever there is partial alignment between $\mathbf{M}$ and $(\mathbf{W}^{\mathrm{bmi}})^\top$, with faster learning as this alignment increases.

Reinforcement learning. To train RNNs with RL, we use a node perturbation learning rule closely related to REINFORCE [6]. We first define a scalar reward $R^t = -|\boldsymbol{\epsilon}^t|^2$. Node perturbation seeks to increase this reward by trial and error using the noise appearing in the population activity in equation (1). If the noise to a given neuron causes the reward to increase, then the weights will be updated so that that neuron's input will be greater in subsequent trials. Including an eligibility trace in the learning rule further makes it possible to assign credit for changes in reward to noise events at previous time steps. Finally, we can subtract off a baseline from the reward so that learning is guided by a reward prediction error, which somewhat decreases the variance of the weight updates without introducing bias [9]. The resulting learning rule, which we derive using the policy gradient theorem [24, 9] in Appendix B.2, is

$$\Delta W_{ij}^{\mathrm{rec}} = \eta \sum_{t}^{T} (R^t - \bar{R}^t) q_{ij}^t, \qquad q_{ij}^t = \left(1 - \frac{1}{\tau}\right) q_{ij}^{t-1} + \frac{1}{\tau}\xi_i^t \phi'(u_i^t) h_j^{t-1} \tag{3}$$

Very similar RL rules for RNNs have been previously proposed and studied in Refs. [25, 8]. Importantly for our purposes, the fact that this learning rule is derived using a policy gradient approach guarantees that, upon averaging over noise, the weight updates follow the true gradient of the loss function (in our case given by $L = -\frac{1}{2}\sum_t R^t$), making this an *unbiased* learning rule. Example output trajectories and the learning curve from an RNN trained on the cursor-control task using (3) are shown in Fig. 2D,E.

Characterizing changes in neural activity with vector flow fields. Because observing changes in neural activity is much more straightforward than observing changes in synaptic weights in neuroscience experiments, we aim to identify experimentally observable signatures in the neural activity that might distinguish biased SL from unbiased RL. Because the RNN activity consists of time-dependent trajectories, and changes in the weights will generally affect entire trajectories cumulatively from one time step to the next, calculating the change $\Delta \mathbf{h}^t$ due to an update $\Delta \mathbf{W}^{\mathrm{rec}}$ is not entirely straightforward (though *cf.* the feedforward case described in Appendix A, where this can be done). A more useful approach is to quantify the learning-induced change in the vector flow field that shapes the RNN activity, where the flow field can be estimated empirically by observing

neural activity trajectories. To do this, we linearize (1), assuming $\phi(u) = u$, to obtain an expression for a vector flow field function $\mathbf{F}(\mathbf{h})$ that maps $\mathbf{h}^t \longrightarrow \mathbf{h}^{t+1}$:

$$\mathbf{F}(\mathbf{h}) = \frac{\mathbf{h}^{t+\Delta t} - \mathbf{h}^t}{\Delta t} = \frac{1}{\tau}\left[(\mathbf{W}^{\mathrm{rec}} - \mathbb{I})\mathbf{h} + \mathbf{W}^{\mathrm{in}}\mathbf{x} + \mathbf{W}^{\mathrm{fb}}\mathbf{y}\right], \tag{4}$$

where, in our simulations, $\Delta t = 1$. If the weights change according to $\mathbf{W}^{\mathrm{rec}} \to \mathbf{W}^{\mathrm{rec}} + \Delta\mathbf{W}^{\mathrm{rec}}$, then the flow field exhibits a corresponding change $\mathbf{F} \to \mathbf{F} + \Delta\mathbf{F}$, where $\Delta\mathbf{F}(\mathbf{h}) = \frac{1}{\tau}\Delta\mathbf{W}^{\mathrm{rec}}\mathbf{h}$. This equation enables us to relate changes in $\mathbf{W}^{\mathrm{rec}}$ to changes in neural activity. As illustrated in Fig. 1D, our approach will be to (i) predict the expected $\Delta\mathbf{F}(\mathbf{h})$ under either SL or RL, (ii) estimate $\Delta\mathbf{F}(\mathbf{h})$ empirically using activity observed early and late in learning, and (iii) determine which learning rule was used to train the network by comparing the observed and predicted flow field changes.

In order to predict the flow field change, we use the noise-averaged learning rules from (2) and (3), which are given in Appendix B.1 and Appendix B.2, respectively. Importantly for our purposes, they depend entirely on quantities that are known or can be estimated from experimental data, including $\mathbf{M}$, $\mathbf{W}^{\mathrm{bmi}}$, $\langle\mathbf{h}^t\rangle$, $\langle\boldsymbol{\epsilon}^t\rangle$, and $\langle\mathbf{h}^t(\mathbf{h}^{t'})^\top\rangle$. By averaging over noise, we effectively assume that the cumulative updates during learning that make up $\langle\Delta\mathbf{W}^{\mathrm{rec}}\rangle$ differ only in having different noise realizations, but are otherwise in a consistent direction. We provide empirical evidence that this is approximately the case in Appendix C. The expected flow field changes assuming either of our two learning rules are then given by $\Delta\mathbf{F}^{\mathrm{pred}}_{\mathrm{SL}}(\mathbf{h}) = \frac{1}{\tau}\langle\Delta\mathbf{W}^{\mathrm{rec}}\rangle|_{\mathrm{SL}}\mathbf{h}$ and $\Delta\mathbf{F}^{\mathrm{pred}}_{\mathrm{RL}}(\mathbf{h}) = \frac{1}{\tau}\langle\Delta\mathbf{W}^{\mathrm{rec}}\rangle|_{\mathrm{RL}}\mathbf{h}$. We can note in particular that, because we are asking only about changes in the flow field rather than the flow field itself, these quantities do not depend explicitly on the recurrent connectivity $\mathbf{W}^{\mathrm{rec}}$ itself, which is presumably unobtainable experimentally.

Next, we wish to estimate the empirical change in the flow field given neural activity observed at different points in learning. The approach is somewhat simplified if we assume that the trials during which learning takes place are preceded and followed by "early" and "late" blocks of trials, in which the performance is observed but the weights are not updated. The assumption that the amount of learning during these early and late blocks is negligible is justified in the limit where the learning rate is small and the number of training trials is large. (Evidence that our main results do not depend strongly on this assumption is given in Appendix A, where learning was allowed to take place during the early and late blocks.) Then, from (4), we define the observed flow field from time step t in trial n as either $\mathbf{F}^{\mathrm{early}}(\mathbf{h}) = \mathbf{A}^{\mathrm{early}}\mathbf{h}$ or $\mathbf{F}^{\mathrm{late}}(\mathbf{h}) = \mathbf{A}^{\mathrm{late}}\mathbf{h}$, where $\mathbf{A}^{\mathrm{early}}$ is the least-squares solution to the autoregression equation $\mathbf{h}^{n,t+1} = \mathbf{A}^{\mathrm{early}}\mathbf{h}^{n,t}$ for all time steps t in all early trials n, and $\mathbf{A}^{\mathrm{late}}$ is the corresponding solution for late trials. The empirical change in the flow field due to learning for any point $\mathbf{h}$ is then given by $\Delta\mathbf{F}^{\mathrm{obs}}(\mathbf{h}) = \mathbf{F}^{\mathrm{late}}(\mathbf{h}) - \mathbf{F}^{\mathrm{early}}(\mathbf{h})$.

Having defined the observed and predicted changes in the flow fields for arbitrary $\mathbf{h}$, we can define the correlation between these quantities as

$$\mathrm{corr}(\Delta\mathbf{F}^{\mathrm{obs}}, \Delta\mathbf{F}^{\mathrm{pred}}) = \frac{1}{N_{\mathrm{trials}}T}\sum_{t=1}^{T}\sum_{n=1}^{N_{\mathrm{trials}}}\frac{\Delta\mathbf{F}^{\mathrm{obs}}(\mathbf{h}^{n,t})\cdot\Delta\mathbf{F}^{\mathrm{pred}}(\mathbf{h}^{n,t})}{|\Delta\mathbf{F}^{\mathrm{obs}}(\mathbf{h}^{n,t})|\,|\Delta\mathbf{F}^{\mathrm{pred}}(\mathbf{h}^{n,t})|}. \tag{5}$$

One important point to note is that, because this quantity is computed only at observed values of $\mathbf{h}^{n,t}$, it is not the case that the flow field needs to be estimated everywhere in the high-dimensional space of neural activity. If the activity tends to occupy a low-dimensional manifold within this space, then the flow field needs to be determined only within this manifold. Another point to note is that this metric is independent of the magnitudes of $\Delta\mathbf{F}^{\mathrm{obs}}$ and $\Delta\mathbf{F}^{\mathrm{pred}}$, so that it does not depend on overall prefactors such as the learning rates and network time constant. In the simulations that follow, the correlation metric (5) will be computed separately under our two hypothesized learning rules, i.e. with $\Delta\mathbf{F}^{\mathrm{pred}} = \Delta\mathbf{F}^{\mathrm{pred}}_{\mathrm{SL}}$ and with $\Delta\mathbf{F}^{\mathrm{pred}} = \Delta\mathbf{F}^{\mathrm{pred}}_{\mathrm{RL}}$. Then the results will be compared in order to infer which of the two learning rules is more likely to have generated the data.

Distinct neural signatures of biased SL and unbiased RL. In order to emulate monkey BMI experiments (e.g., [10, 13, 12, 11]), we first pretrain the recurrent weights $\mathbf{W}^{\mathrm{rec}}$ of an RNN with fixed, random readout weights $\mathbf{W}^{\mathrm{bmi0}}$ to perform a center-out cursor-control task, in which a cursor must be moved to one of four target locations, specified by the input to the RNN. This results in

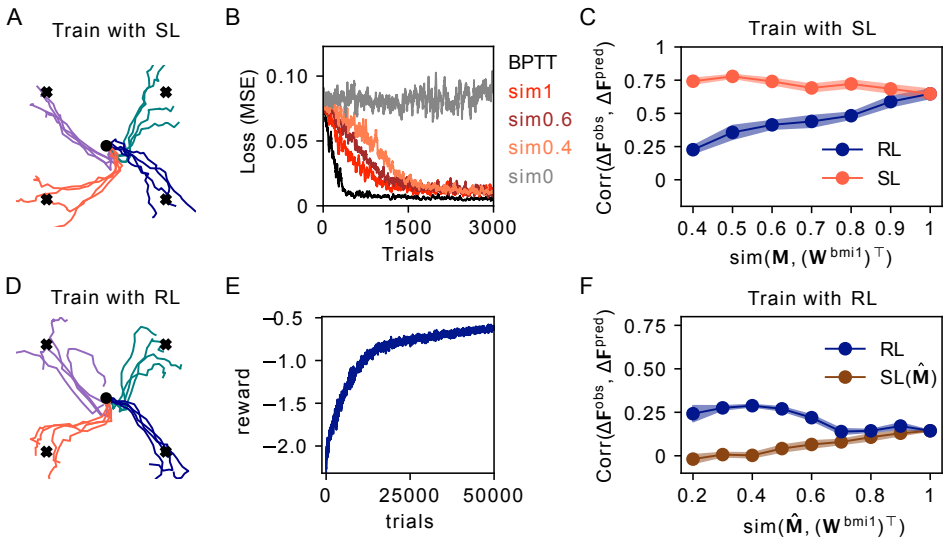

Figure 2: Distinguishing learning rules in a trained network. (A) Example output traces from an RNN trained with SL to move a cursor to one of four targets. (B) When trained with SL, the similarity of the credit assignment mapping with the readout weights determines the speed of training (red curves). Learning in an RNN trained with backpropagation through time is shown for comparison (black curve). (C) For an RNN trained with SL, the similarity of the observed flow field change with the predicted change assuming either SL (red) or RL (blue). (D) Example output traces from an RNN trained with RL to move a cursor to one of four targets. (E) Learning curve for an RNN trained with RL. (F) For an RNN trained with RL, the similarity of the observed flow field change with the predicted change assuming either SL with a randomly sampled $\hat{\mathbf{M}}$ (brown) or RL (blue).

low dimensional recurrent dynamics, analogous to the "intrinsic manifold" of activity observed in neural recordings [10, 19]. We then change the BMI decoder to $\mathbf{W}^{\text{bmi1}}$, make identical copies of the network, and train one with SL and the other with RL. For these simulations, shown in Fig. 2, we make the simplifying assumptions (to be generalized in later sections) that (i) the credit assignment mapping $\mathbf{M}$ is fixed and known; (ii) the feedback weights $\mathbf{W}^{\text{fb}}$ are zero; and (iii) the noise $\boldsymbol{\xi}^t \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is isotropic. We show in Appendix C that the results shown in Fig. 2 are robust to changes in hyperparameters.

For the networks trained with biased SL, we find that the comparing the observed and predicted flow field changes correctly detects the bias (Fig. 2C). After pretraining, we change the readout weights to $\mathbf{W}^{\text{bmi1}}$ such that the alignment (defined here as the cosine similarity of the flattened matrices) between the decoders is $\text{sim}(\mathbf{W}^{\text{bmi0}}, \mathbf{W}^{\text{bmi1}}) = 0.5$. We next train networks with SL using a credit assignment mapping $\mathbf{M}$, then calculate the correlation between the change in flow field from the simulated data and the predicted change in flow field based on either the RL prediction (which implicitly depends on $\mathbf{W}^{\text{bmi1}}$) or the SL prediction (which depends on $\mathbf{M}$). As we vary the alignment between $\mathbf{M}$ and $(\mathbf{W}^{\text{bmi1}})^\top$, we find that the correlation metric correctly detects the bias present during learning whenever these two matrices are sufficiently distinct (e.g. $\text{sim}(\mathbf{M}, (\mathbf{W}^{\text{bmi1}})^\top) < 0.9$). For the networks trained with RL using the same procedure described above, we find that the change in activity is aligned more closely to the prediction using true decoder $\mathbf{W}^{\text{bmi1}}$ than when using a random matrix $\hat{\mathbf{M}}$ that is partially aligned to $\mathbf{W}^{\text{bmi1}}$. As we increase the alignment between $\text{sim}(\mathbf{W}^{\text{bmi0}}, \mathbf{W}^{\text{bmi1}})$, this distinction is less discernible (Fig. 2F).

In the empirical results that follow, we will generalize the assumptions made thus far and show that there is a significant range of parameters where the results still hold. In particular, we will consider the roles played by (i) estimation and/or learning of the credit-assignment model $\mathbf{M}$, (ii) driving feedback via the weights $\mathbf{W}^{\text{fb}}$, and (iii) non-isotropic noise.

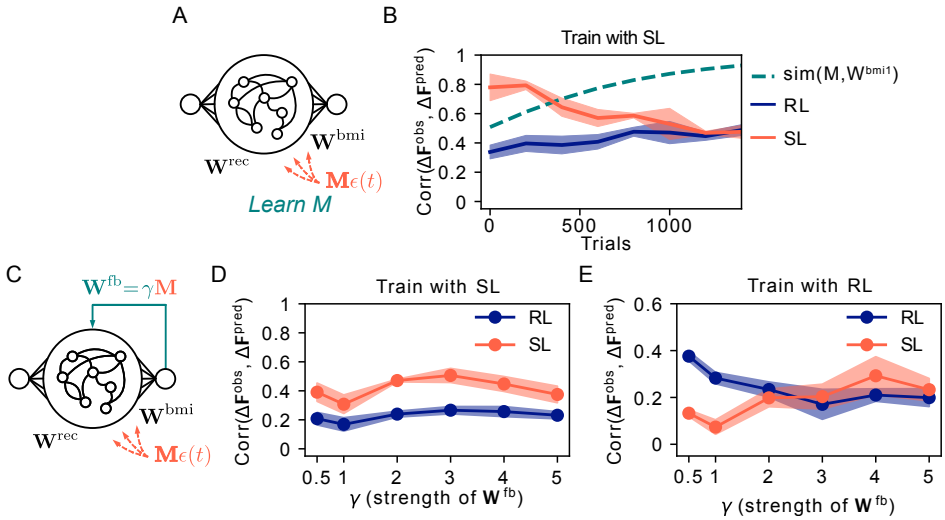

Figure 3: Learning the credit assignment mapping and driving with visual feedback. (A) In an RNN trained with SL, the credit assignment mapping $\mathbf{M}$ is learned online during training. (B) The similarity of $\mathbf{M}$ to $(\mathbf{W}^{\mathrm{bmi}})^\top$ increases with training due to learning of $\mathbf{M}$ (green). The flow field metric correctly identifies SL as the learning rule early in training when $\mathbf{M}$ and $\mathbf{W}^{\mathrm{bmi}}$ are dissimilar, but not later in training when they are very similar. (C) An RNN (now with fixed $\mathbf{M}$) is additionally driven by a feedback signal via weights $\mathbf{W}^{\mathrm{fb}} \propto \mathbf{M}$. (D) Comparison between predicted and observed changes in the activity flow field as a function of feedback gain for an RNN trained with SL. (E) Comparison between predicted and observed changes in the activity flow field as a function of feedback gain for an RNN trained with SL.

Learning the credit assignment mapping. In order to optimize learning in a supervised setting, a learner's credit assignment mapping could improve over time to align more closely with the decoder (Fig. 3A). To investigate the effect of such learning in our model, we applied our analysis to an RNN in which recurrent weights $\mathbf{W}^{\mathrm{rec}}$ were trained with SL, with a biased credit assignment mapping $\mathbf{M}$ that slowly improved via *weight mirroring*, a perturbation-based algorithm for learning a "backward-pass" credit assignment matrix [14, 15]. In our simulations (Fig. 3B), the alignment between $\mathbf{M}$ and $(\mathbf{W}^{\mathrm{bmi1}})^\top$ is initially set at 0.5, then slowly increases as $\mathbf{M}$ is learned simultaneously along with the recurrent weights $\mathbf{W}^{\mathrm{rec}}$. Instead of predicting the change in flow field before and after learning, we now apply our analysis over discrete windows during learning. As expected, activity changes are biased—and hence the SL and RL learning rules are distinguishable—early in learning. However, these differences disappear as $\mathbf{M}$ approaches $(\mathbf{W}^{\mathrm{bmi1}})^\top$ later in learning.

Incorporating driving feedback. So far, we have not included driving feedback, which can strongly influence the dynamics of population activity. We next included fixed driving feedback weights $\mathbf{W}^{\mathrm{fb}} = \gamma\mathbf{M}$ and varied the strength of the driving feedback weights by a scalar factor γ (Fig. 3C). We found that bias in an RNN trained with SL is correctly identified for a wide range of γ (Fig. 3D). For RNNs trained with RL, we found that the correct learning rule is identified for weak feedback, but that the metric fails for strong feedback gain in this case (Fig. 3E), likely because neural activity in this limit is dominated by the contribution from $\mathbf{W}^{\mathrm{fb}}$, which is proportional to $\mathbf{M}$.

Estimating the credit assignment mapping from observed data. So far, we have estimated the extent to which the activity updates follow either the direction of the gradient defined by the decoder weights or the (possibly biased) gradient defined by the credit assignment mapping. While this is of interest in and of itself, it is not immediately obvious how an experimenter might go about estimating the credit assignment mapping $\mathbf{M}$.

To obtain an estimate $\hat{\mathbf{M}}$ of $\mathbf{M}$, we follow an approach previously used in experiments, in which dimensionality reduction is first applied to the neural data, followed by linear regression to relate the network activity to the cursor movements [10, 13, 19]. More specifically, we (i) train the RNN

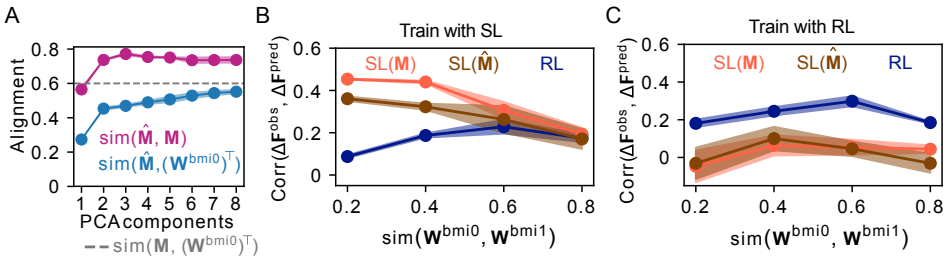

Figure 4: Estimating the credit assignment mapping. (A) The similarities between the estimated $\hat{\mathbf{M}}$ and the true $\mathbf{M}$ (magenta) and between $\hat{\mathbf{M}}$ and $(\mathbf{W}^{\text{bmi0}})^\top$ (blue) as a function of the number of principal components used in the regression in an RNN trained with SL. (B) Comparison between predicted and observed changes in the activity flow field as a function of old vs. new decoder alignment for an RNN trained with SL. Model predictions assume SL using the true credit assignment mapping $\mathbf{M}$ (red), SL using the estimated $\hat{\mathbf{M}}$ (brown), or RL (blue). (C) Same as (B), for an RNN trained with RL.

with decoder $\mathbf{W}^{\text{bmi0}}$ and biased credit assignment mapping $\mathbf{M}$ (where $\text{sim}(\mathbf{M}, \mathbf{W}^{\text{bmi0}}) = 0.6$), (ii) observe the activity and cursor trajectories produced by the trained network, (iii) perform principal components analysis on the network activity and relate the principal components to the cursor positions with linear regression, and (iv) use the regression coefficients to define the estimated internal model $\hat{\mathbf{M}} = \mathbf{DC}$, where $\mathbf{C}$ is the matrix of principal-component eigenvectors and $\mathbf{D}$ is the matrix of regression coefficients. Fig. 4A shows that the $\hat{\mathbf{M}}$ obtained in this way is similar to the true $\mathbf{M}$ used to train the RNN. Important to note for our aim of distinguishing learning rules is that $\hat{\mathbf{M}}$ is more similar to $\mathbf{M}$ than to $\mathbf{W}^{\text{bmi0}}$. We also see from Fig. 4A that this result depends only weakly on the number of principal components used to describe the network activity.

After estimating $\hat{\mathbf{M}}$, we changed the decoder to $\mathbf{W}^{\text{bmi1}}$ and trained the network to proficiency with either SL (using the fixed $\mathbf{M}$ in the learning update) or RL. For an RNN trained with SL (Fig. 4B), the correct learning rule is identified using either the true $\mathbf{M}$ (red curve) or the estimated $\hat{\mathbf{M}}$ (brown curve) for low and intermediate degrees of similarity between $\mathbf{W}^{\text{bmi0}}$ and $\mathbf{W}^{\text{bmi1}}$. For an RNN trained with RL (Fig. 4C), correct identification of the learning rule does not depend on the similarity between $\mathbf{W}^{\text{bmi0}}$ and $\mathbf{W}^{\text{bmi1}}$. Together, the above results show that, for a relatively wide range of parameters, the credit assignment mapping can be estimated from observed data, and that this estimate is sufficient to distinguish between learning rules used to train different networks.

RL with non-isotropic noise. In the theoretical and empirical results above, we have assumed that the noise covariance is isotropic. If we instead allow for non-isotropic noise covariance $\mathbf{\Sigma}$, we find that this covariance appears in the expected RL weight update (Appendix B.2). As in feedforward networks [26], this introduces a *bias* in the weight updates, so that the noise-averaged weight updates no longer follow the true gradient. Given that RL is no longer unbiased in this case, we thus asked under what conditions we might still be able to distinguish biased SL from RL.

We first established that the RNNs can learn a cursor-control task with non-isotropic noise via either SL (trivially) or RL (as long as some of the noise is in the subspace of the decoder). To do this, we pretrained networks with $\mathbf{W}^{\text{bmi0}}$, changed the decoder to $\mathbf{W}^{\text{bmi1}}$, and then trained using either SL or RL. In the SL case, we chose a biased credit mapping $\mathbf{M}$ that has partial overlap with the new decoder, with $\text{sim}(\mathbf{M}, (\mathbf{W}^{\text{bmi1}})^\top) = 0.6$. The noise covariance was chosen to be d-dimensional, with $\text{rank}(\mathbf{\Sigma}) = d$, and isotropic within those dimensions. The first two of these dimensions were selected to lie in the subspace spanned by $\mathbf{M}$ for the SL-trained RNNs or by $\mathbf{W}^{\text{bmi1}}$ for the RL-trained RNNs. Other components of $\mathbf{\Sigma}$ added in random dimensions orthogonal to this subspace and to one another.

The results in Fig. 5 show that, as expected, we cannot reliably distinguish the learning rule used to train the network when the noise is very low-dimensional, either when training with SL (Fig. 5B) or with RL (Fig. 5C). As the noise dimensionality increases, however, the learning rules become more

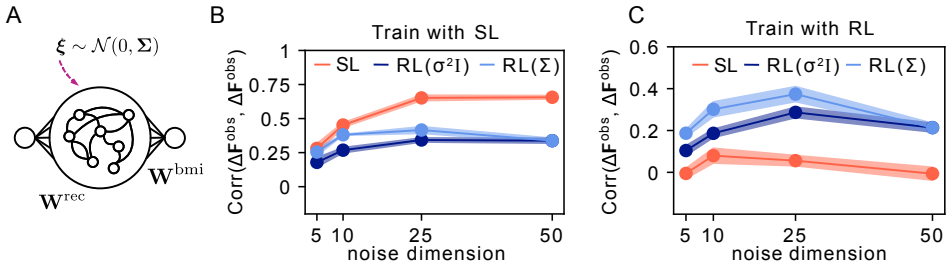

Figure 5: Distinguishing learning rules with non-isotropic noise. (A) Correlated noise with covariance matrix $\mathbf{\Sigma}$ is injected into the network during training and testing. (B) For RNNs trained with SL, the similarities between the observed flow field change and the predicted flow field change assuming SL (red), RL using the true noise covariance (light blue), or RL using a naive isotropic estimate of the noise covariance (dark blue). (C) Same as (B), but for RNNs trained with RL.

easily distinguishable in both cases. This remains true even if we do not presume to know the true noise covariance matrix, but instead use a naive assumption of isotropic noise in order to generate the predictions (dark blue lines in Fig. 5B,C).

3 Discussion

In this work, we have proposed a method for distinguishing biased SL from unbiased RL under the assumption that the mapping from brain to behavior is known, as in a BMI experiment. Both SL and RL are plausible candidates for how learning might learn in the brain. Cerebellum in particular has been hypothesized as a classic system for SL, both in classification and continuous motor-control settings [27, 28, 29], while basal ganglia are classically viewed as the seat of RL in the brain [30, 31]. These areas are both part of the brain's motor-control circuitry, connected by various direct and indirect pathways to the motor cortex, where BMI experiments are most often performed.

One possible limitation of our approach is that the number of learning rules that might possibly be implemented in recurrent circuits in the brain is much greater than the two that we have focused on here. Nevertheless, we view the SL and RL rules that we consider here as representatives of two fundamental classes of learning rules: one in which learning is based on a model for credit assignment, where this model is necessarily imperfect, and another which uses no such model. While we cannot test every learning algorithm from these classes, we conjecture that the basic fact that allows us to distinguish our SL and RL rules—namely, that the expected weight update is in the image of the credit-assignment matrix for model-based rules but in the image of the true readout for non-model-based rules—is a fairly generic feature of these two classes. To support this, we show in Appendix C that our main results hold for different versions of the biased and unbiased learning rules that we have studied.

While we have shown that SL and RL are in principle distinguishable in simulation, there will be challenges in extending this analysis to real data. For example, reliable estimation of our flow field correlation metric might require large populations of neurons recorded for long periods of time. Other potential challenges that will need to be addressed include accounting for the possibility that multiple types of learning might be occurring simultaneously at different locations, as well as considering cognitive mechanisms such as attention and contextual inference, which we have not included in our model. Finally, there are possible ambiguities to consider in mapping our model onto the brain's motor circuitry. While the most straightforward interpretation is that the RNN in our model corresponds to motor cortex, a more elaborate interpretation could be made by noting that both the cerebellum and the basal ganglia have well-established roles in motor learning [32] and are likely to play roles in BMI learning as well [31, 33].

References

[1] Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63, 1987.

[2] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

[3] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[4] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.

[5] James M Murray. Local online learning in recurrent networks with random feedback. *eLife*, 8: e43299, 2019.

[6] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[7] Justin Werfel, Xiaohui Xie, and H Seung. Learning curves for stochastic gradient descent in linear feedforward networks. *Advances in neural information processing systems*, 16, 2003.

[8] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899, 2017.

[9] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[10] Patrick T Sadtler, Kristin M Quick, Matthew D Golub, Steven M Chase, Stephen I Ryu, Elizabeth C Tyler-Kabara, M Yu Byron, and Aaron P Batista. Neural constraints on learning. *Nature*, 512(7515):423–426, 2014.

[11] Xiao Zhou, Rex N Tien, Sadhana Ravikumar, and Steven M Chase. Distinct types of neural reorganization during long-term learning. *Journal of neurophysiology*, 121(4):1329–1341, 2019.

[12] Vivek R Athalye, Karunesh Ganguly, Rui M Costa, and Jose M Carmena. Emergence of coordinated neural dynamics underlies neuroprosthetic learning and skillful control. *Neuron*, 93 (4):955–970, 2017.

[13] Emily R Oby, Matthew D Golub, Jay A Hennig, Alan D Degenhart, Elizabeth C Tyler-Kabara, M Yu Byron, Steven M Chase, and Aaron P Batista. New neural activity patterns emerge with long-term learning. *Proceedings of the National Academy of Sciences*, 116(30):15210–15215, 2019.

[14] Mohamed Akrout, Collin Wilson, Peter C Humphreys, Timothy Lillicrap, and Douglas Tweed. Deep learning without weight transport. *arXiv preprint arXiv:1904.05391*, 2019.

[15] Benjamin James Lansdell, Prashanth Ravi Prakash, and Konrad Paul Kording. Learning to solve the credit assignment problem. In *International Conference on Learning Representations*, 2019.

[16] Sukbin Lim, Jillian L McKee, Luke Woloszyn, Yali Amit, David J Freedman, David L Sheinberg, and Nicolas Brunel. Inferring learning rules from distributions of firing rates in cortical neurons. *Nature neuroscience*, 18(12):1804–1810, 2015.

[17] Aran Nayebi, Sanjana Srivastava, Surya Ganguli, and Daniel L Yamins. Identifying learning rules from neural network observables. *Advances in Neural Information Processing Systems*, 33:2639–2650, 2020.

[18] Daniel R Kepple, Rainer Engelken, and Kanaka Rajan. Curriculum learning as a tool to uncover learning principles in the brain. In *International Conference on Learning Representations*, 2021.

[19] Barbara Feulner and Claudia Clopath. Neural manifold under plasticity in a goal driven learning behaviour. *PLoS computational biology*, 17(2):e1008621, 2021.

[20] Barbara Feulner, Matthew G Perich, Raeed H Chowdhury, Lee E Miller, Juan A Gallego, and Claudia Clopath. Small, correlated changes in synaptic connectivity may facilitate rapid motor learning. *bioRxiv*, 2021.

[21] Matthew D Golub, Patrick T Sadtler, Emily R Oby, Kristin M Quick, Stephen I Ryu, Elizabeth C Tyler-Kabara, Aaron P Batista, Steven M Chase, and M Yu Byron. Learning by neural reassociation. *Nature neuroscience*, 21(4):607–616, 2018.

[22] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[23] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[24] Peter Marbach and John N Tsitsiklis. Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.

[25] Ila R Fiete and H Sebastian Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical review letters*, 97(4):048104, 2006.

[26] Daniel N. Scott and Michael J. Frank. Beyond gradients: Noise correlations control hebbian plasticity to shape credit assignment. *bioRxiv*, 2021. doi: 10.1101/2021.11.19.466943.

[27] James S Albus. A theory of cerebellar function. *Mathematical biosciences*, 10(1-2):25–61, 1971.

[28] John Porrill, Paul Dean, and Sean R Anderson. Adaptive filters and internal models: multilevel description of cerebellar function. *Neural Networks*, 47:134–149, 2013.

[29] Jennifer L Raymond and Javier F Medina. Computational principles of supervised learning in the cerebellum. *Annual review of neuroscience*, 41:233–253, 2018.

[30] Peter Redgrave, Tony J Prescott, and Kevin Gurney. The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, 89(4):1009–1023, 1999.

[31] Vivek R Athalye, Jose M Carmena, and Rui M Costa. Neural reinforcement: re-entering and refining neural dynamics leading to desirable outcomes. *Current opinion in neurobiology*, 60: 145–154, 2020.

[32] John W Krakauer, Alkis M Hadjiosif, Jing Xu, Aaron L Wong, and Adrian M Haith. Motor learning. *Compr Physiol*, 9(2):613–663, 2019.

[33] Jay A Hennig, Emily R Oby, Darby M Losey, Aaron P Batista, M Yu Byron, and Steven M Chase. How learning unfolds in the brain: toward an optimization view. *Neuron*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See, the Discussion
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We emphasize important assumptions in the main text, and derive more extensive proofs in the Appendix
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs are included in the Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code included as supplementary material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] All simulations were run on a CPU
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]