Neural Networks 153 (2022) 349–372

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Computational role of exploration noise in error-based de novo motor learning

Lucas Rebelo Dal'Bello [a], Jun Izawa [b,*]

[a] School of Integrative and Global Majors, University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan
[b] Faculty of Engineering, Information and Systems, University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan

ARTICLE INFO

Article history:
Received 17 December 2021
Received in revised form 23 April 2022
Accepted 9 June 2022
Available online 16 June 2022

Keywords:
Motor learning
Exploration
Sensitivity derivative
Computational model
Redundancy
Error-based learning

ABSTRACT

The redundancy inherent to the human body is a central problem that must be solved by the brain when acquiring new motor skills. The problem of redundancy becomes particularly critical when learning a new motor policy from scratch in a novel environment and task (i.e., de novo learning). It has been proposed that motor variability could be leveraged to explore and identify task-potent motor commands, and recent results indicated a possible role of motor exploration in error-based motor learning, including in de novo learning tasks. However, the precise computational mechanisms underlying this role remain poorly understood. A new controller in a de novo motor task can potentially be learned by first using motor exploration to learn a sensitivity derivative, which can transform observed task errors into motor corrections, enabling the error-based learning of the controller. Although this approach has been discussed, the computational properties of exploration and how this mechanism can explain recent reports of motor exploration in error-based de-novo learning have not been thoroughly examined. Here, we used this approach to simulate the tasks used in several recent studies of human motor learning tasks in which motor exploration was observed, and replicating their main results. Analyses of the proposed learning mechanism using equations and simulations suggested that exploring the entire motor command space leads to the training of an efficient sensitivity derivative, enabling rapid learning of the controller, in visuomotor adaptation and de novo tasks. The successful replication of previous experimental results elucidated the role of motor exploration in motor learning.

1. Introduction

Coordinating the various degrees of freedom of the human body during accurate and precise movements is a fundamental problem of motor learning (Bernstein, 1967). Because of the redundancy of the degrees of freedom of the human body, movements can be realized with an infinite number of coordination strategies, of which only one is selected and executed by the motor system. This redundancy is present at many levels of the motor system, ranging from joints (Dal'Bello & Izawa, 2021; Mosier et al., 2005; Singh et al., 2016) to muscles (Berger et al., 2013; Hirashima & Oya, 2016) and to the neurons that are responsible for movement (Hennig et al., 2018; Sadtler et al., 2014). However, the mechanism by which this redundancy is solved remains largely unknown.

The problem of redundancy is at the core of the acquisition of new motor skills, a process also known as de novo learning (Krakauer et al., 2019; Sternad, 2018; Telgen et al., 2014). Acquiring a new motor skill requires the learning of a new control policy from scratch, reflecting learned information about the arbitrary relationships between redundant motor actions and outcomes in a novel task. Brain–machine interfaces (BMIs) also require the learning of the redundant relationship between neural activation and outcomes, which is not fixed due to the high variability in neural recording conditions within- and across-days (Sussillo et al., 2016). It has been suggested that the basal ganglia play an important role in this process (Gutierrez-Garralda et al., 2013; Krakauer et al., 2019).

To solve the problem of redundancy inherent to the human body, it has been proposed that, over the course of learning, motor variability is shaped along with the learned structure of the task, so that variability is reduced in the degrees of freedom that affect task outcome (Sternad, 2018). Motor exploration is used, potentially through a reinforcement learning process (Sutton & Barto, 2018), to identify a "solution manifold" in the motor command space, with a progressive increase of usage of more rewarding motor commands (Haith & Krakauer, 2013). In line with this hypothesis, there is evidence that motor variability is correlated with reward-based learning (Dhawale et al., 2017; Izawa

* Corresponding author.
E-mail addresses: lucasrdb@hebbs.emp.tsukuba.ac.jp (L.R. Dal'Bello), izawa@emp.tsukuba.ac.jp (J. Izawa).

https://doi.org/10.1016/j.neunet.2022.06.011

& Shadmehr, 2011; Pekny et al., 2015; Wu et al., 2014), with studies in songbirds suggesting that variability-induced learning is regulated by circuits that are homologous to cortico-basal ganglia circuits in mammals (Ölveczky et al., 2005).

It was recently reported that motor variability might also play a role in error-based motor adaptation, with variability exhibiting a correlation with the amount of adaptation in error-based motor tasks in humans (Wu et al., 2014), including in task-irrelevant dimensions, when redundancy was accounted for (Singh et al., 2016). Recently, when the reward-based and error-based components of adaptation to single-trial perturbations were dissociated, variability in both task-relevant and task-irrelevant dimensions was shown to be correlated with error-based adaptation (Dal'Bello & Izawa, 2021). However, the precise role of motor exploration in error-based motor adaptation in a redundant biological system is still unclear. Thus, the aim of the current study was to provide a computational account of recently reported roles of exploration during error-based motor adaptation (Singh et al., 2016) and de novo learning (Berger et al., 2013; Dal'Bello & Izawa, 2021).

Computationally, to train a motor policy by the experienced visual error, the error in the task domain should be transformed back to the motor commands space, a process thought to rely on an internal model, also called a sensitivity derivative (Abdelghani et al., 2008; Hadjiosif et al., 2021). In this framework, a sensitivity derivative is initially unknown in a de novo task, and its learning, possibly via an exploration-driven process, can then drive the error-based learning of a new controller for the task. Although the notion of having to learn a sensitivity derivative before learning a controller has been discussed in the past (Jordan & Rumelhart, 1992; Pierella et al., 2019), it has not been investigated in the context of a possible role of motor exploration on motor learning (Dal'Bello & Izawa, 2021; Singh et al., 2016; Wu et al., 2014).

In the current study, we examined this possible mechanism with simulations of artificial neural networks learning various redundant motor control problems. In our simulations, sensitivity derivatives represented by internal models were trained under various amounts of exploration in different subspaces of motor commands and used to modify control policies represented by inverse models, with the goal of observing the effects of exploration on learning of both sensitivity derivatives and control policies. By controlling the exploration variability and the initial conditions of the many simulations, we aimed to replicate the main results of several papers on motor learning in which we believe motor exploration played a key role (Berger et al., 2013; Dal'Bello & Izawa, 2021; Singh et al., 2016), to test the validity of the proposed mechanism as a major component of de novo learning.

2. Materials and methods

2.1. General framework

We employed a similar framework in all of our simulations, making parallels between the regions in the brain responsible for motor execution and adaptation (Fig. 1) and the specific functions involved in our proposed learning mechanism (Fig. 2). In our framework, intended movements, possibly represented in the posterior parietal cortex (PPC) (Desmurget et al., 2009) as extrinsic parameters, are transformed into muscle-centered, intrinsic parameters in the motor cortex (Kalaska, 2009) through an internal inverse model, and then are delivered to the spinal cord and finally to the muscles that execute the intended movement (Fig. 1). This feedforward pathway going from neurons in the motor cortex to the final movement can be called the environment of the motor task to be controlled, and due to the redundancy

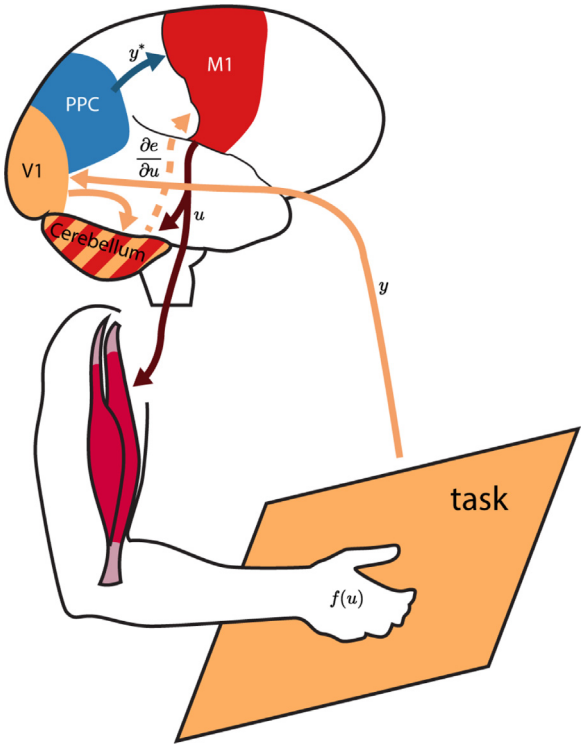

Fig. 1. Schematic of brain regions possibly involved in the proposed error-based learning mechanism. The PPC region generates signals encoding desired outcomes $\boldsymbol{y}^*$, in extrinsic coordinates, which are relayed to M1 (red). M1 encodes the desired outcomes into a motor command signal $\boldsymbol{u}$ in intrinsic coordinates (in our framework, the motor command $\boldsymbol{u}$ is generated from the inverse model g acting on the desired outcome $\boldsymbol{y}^*$). The motor commands are then relayed to muscles via the spinal cord, generating movements and realizing a task outcome $\boldsymbol{y}$ (green) (in our framework, generated from the environment f and the motor command $\boldsymbol{u}$). The task outcome is observed through vision and represented by an outcome signal in V1 (green), in extrinsic coordinates, which is relayed to the cerebellum (greed/red), which, along with an efferent copy of the motor command signal, calculates an error signal that is relayed back to M1 via the thalamus to update the generated motor commands to reduce the error (in our framework, the task error $\boldsymbol{e}$ is transformed via a motor correction $\Delta\boldsymbol{u}$, in intrinsic coordinates, by the sensitivity derivative $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{u}}$). PPC: posterior parietal cortex. M1: primary motor cortex. V1: primary visual cortex. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of this environment (with multiple neurons being responsible for the task outcome), the motor command space in the motor cortex contains dimensions which are task-potent (patterns of neural activations which directly affect the task outcome) and task-null (neural activations which do not affect the task outcome). After the movement execution, the task outcome is then visualized and represented as extrinsic parameters at the visual cortex. This visual information is combined with a predicted task outcome calculated by an internal forward model at the cerebellum (Shadmehr et al., 2010) to form a sensory prediction error signal (Tanaka et al., 2020), in extrinsic parameters. This error signal is finally transformed into an intrinsic parameter signal through a sensitivity derivative (Abdelghani et al., 2008), possibly in the cerebello–thalamo–cortical pathway (Aumann, 2002), and relayed back to the motor cortex (Tseng et al., 2007), where it alters the synaptic connectivity of the cortex (the inverse model), resulting in a reduction in task error when the same intended movement is executed again. During skilled motor control (Fig. 2A), this sensitivity derivative can appropriately correct eventual task errors, with proper modifications of task-potent patterns of neural activations in the motor cortex.

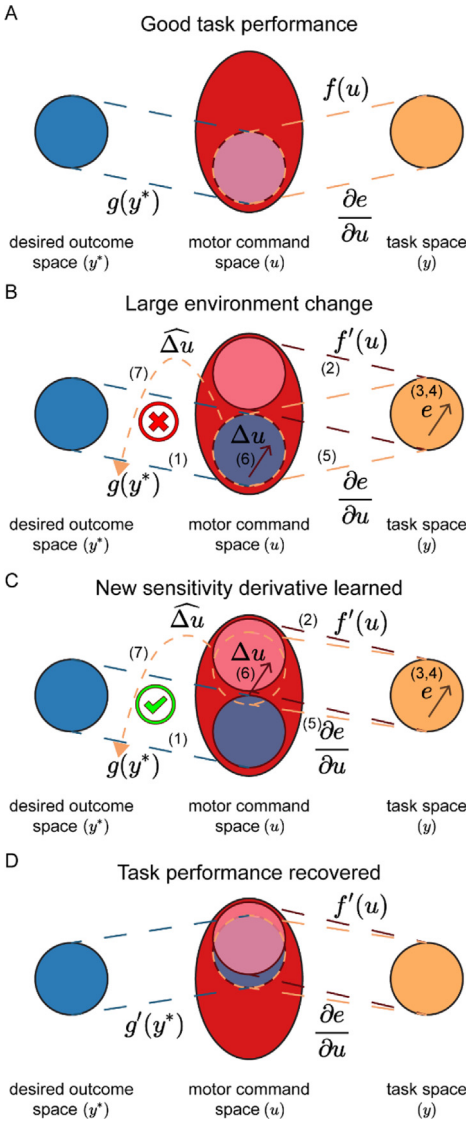

Fig. 2. Schematic of the elements involved in the generation and update of motor commands at different stages of de novo learning. An inverse model g transforms desired outcome signals $\boldsymbol{y^*}$ (left, blue) into motor commands $\boldsymbol{u}$ (middle, red), which are further transformed in a task outcome signal $\boldsymbol{y}$ (right, green) by the environment f. If the task outcome contains an error, this error is transformed into a motor update signal $\Delta\boldsymbol{u}$ by the sensitivity derivative $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{u}}$, which is then used to update the inverse model to reduce the error. Colors represent brain areas in which the encoding of the neural information possibly occurs, in Fig. 1. (A) During baseline control, the inverse model generates appropriate motor commands which will realize the correct task outcome. Eventual errors are transformed into appropriate motor update signals by the sensitivity derivative. (B) When the task environment suffers a large change (from f to f'), the relationship between motor commands and task outcomes changes (motor commands which previously affected the task outcome now do not change the outcome), and the (untrained) inverse model can no longer generate appropriate motor commands. Although there are error signals available, the (untrained) sensitivity derivative generates motor corrections which were appropriate for the previous environment, but are not appropriate for the current environment, and the update of the inverse model cannot proceed. (C) By using motor variability to explore different motor commands, the motor system can update its sensitivity derivative to generate motor corrections appropriate for the current environment, enabling the update of the inverse model. (D) Using the updated sensitivity derivative, the inverse model can be updated (from g to g') to generate motor commands with no task error once again. The exact solution the inverse model converges to depends on the accuracy of the sensitivity derivative. Numbers (1) through (7) in panels B and C correspond to the processes described in the General Framework subsection of the Materials and Methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

When the environment of the motor task exhibits a large change (Fig. 2B), the subspace of task-potent motor commands changes, and the motor commands generated by the inverse model no longer lead to the desired outcomes in the task. Moreover, because the relationship between motor commands and task outcomes has changed, the inverse relationship, between task errors and motor corrections, also changes, and the sensitivity derivative does not generate appropriate motor correction signals that drive the update of the inverse model. We believe that this large change in the environment, requiring the relearning of the sensitivity derivative, is equivalent to a de novo learning task, which has been characterized by the need to learn from scratch a new controller for the task, instead of adapting an existing controller (Krakauer et al., 2019). Tasks which could be described as de novo learning tasks include arm reaching under mirror reversal (Hadjiosif et al., 2021; Telgen et al., 2014; Yang et al., 2021), novel body–machine interfaces such as arbitrary mappings between hand gestures/movements and cursor direction on a screen (Dal'Bello & Izawa, 2021; Liu et al., 2011; Mosier et al., 2005; Rohde et al., 2019; Thorp et al., 2017), virtual "surgeries" that change the force direction (in a virtual environment) of muscles in the arm (Barradas et al., 2020; Berger et al., 2013), and BMIs (Sadtler et al., 2014).

Under our proposed framework, error-based, and not reward-based, de novo learning is involved in the learning of both sensitivity derivatives (in our simulations, represented by forward models) and inverse models. Specifically, the brain explores the motor command space by adding noise to motor commands generated by the inverse model, then observes the outcomes of the noisy motor commands in the (new) task space, and uses the motor command-task outcome pairs as training samples for the sensitivity derivative. Once the sensitivity derivative is sufficiently trained (Fig. 2C), it can be used by the brain to generate motor correction signals appropriate for the new environment, and when used to train the inverse model, it leads to a decrease in errors, enabling skilled motor control once more (Fig. 2D).

An important assumption in our framework is that the motor correction signal generated by the sensitivity derivative is limited in its amplitude. We found that the addition of this feature enabled the framework to reproduce the effects of task-null exploration on motor learning. Such a limitation could possibly be implemented in the brain through biological constraints regarding how the motor correction signal is encoded. The stimulation of neural correlates of errors in the motor cortices have been shown to increase the error in the direction opposite to their preferred direction, suggesting that these neurons encode motor corrections (Inoue et al., 2016). Because of the encoding of such motor correction via the firing rate of a population of neurons (Dayan & Abbott, 2001), it is reasonable to assume that biophysical constraints of neurons, such as a limit in firing rates, can limit the amplitude of an encoded motor correction signal. In our simulations, we implemented such limitations by setting a fixed maximum norm for the motor correction vector generated by the sensitivity derivative, so that if the norm of the vector exceeded this maximum norm, its norm would be rescaled to the maximum norm.

The general flow of how inverse and forward models are trained in our simulations is represented in Fig. 3. In general, one epoch of training of the inverse model in our simulations (Fig. 3A) consisted of: (1) forward pass with a desired outcome through the inverse model; (2) forward pass with the motor command generated by the inverse model through both the environment and the forward model; (3) perturbation of the task outcome, if a perturbation is added; (4) computation of the error between the task outcome and the desired outcome; (5) backward pass with the error through the forward model, back to its inputs (similar to

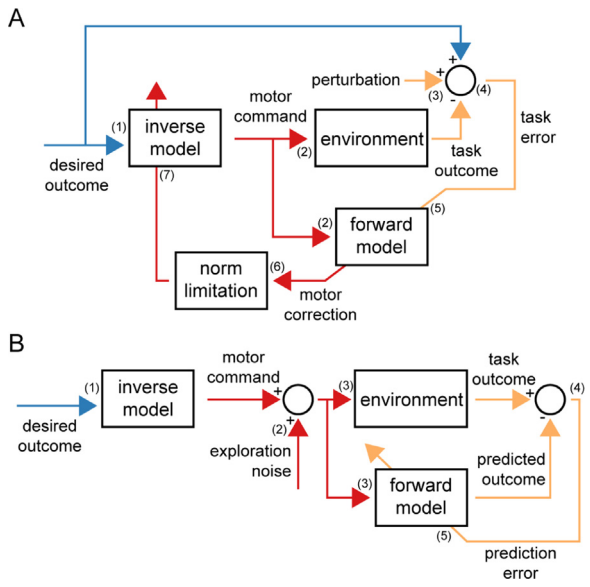

Fig. 3. Flowchart of the update of the inverse and forward models in our simulations. (A) Update of the inverse model. (B) Update of the forward model. Colors represent brain areas in which the encoding of the neural information possibly occurs, in Figs. 1 and 2. Numbers (1) through (7) in panel A and (1) through (5) in panel B correspond to the processes described in the General Framework subsection of the Materials and Methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

what was proposed in Jordan & Rumelhart, 1992); (6) limitation of the norm of the motor correction signal; and (7) backward pass with the computed motor correction through the inverse model, modifying its weights with a fixed learning rate. These numbered steps are shown in Fig. 2 as well.

One epoch of training of the forward model (Fig. 3B) starts similarly, but has a few key differences: (1) a forward pass with the desired outcome through the inverse model; (2) exploration noise is added to the motor command generated by the inverse model; (3) forward pass with the motor command through both the environment and the forward model; (4) computation of the prediction error between the task outcome and the predicted task outcome; (5) backward pass with the prediction error through the forward model, modifying its weights.

We first show the equations and simulation results of a simple learning system to demonstrate the features of our framework, such as how the amount and shape of the motor noise influences the training of the forward model and its subsequent usage as a sensitivity derivative. Specifically, we show how the exploration of the entire motor command space is necessary if the goal is to train a sensitivity derivative that will generate an ideal motor correction. We then report the results from simulations of the tasks reported in three recent studies, whose results we believe are well explained by our proposed framework. Finally, we discuss the implications of our framework in these simulated tasks and in other de novo learning motor tasks.

2.2. Simple linear motor task

2.2.1. Definition of the task

Let us define an n dimensional task space and an m dimensional motor command space, with $m > n$. We define an environment function f, which maps motor commands onto task outcomes, and is represented by an environment matrix $\boldsymbol{M}$, $n \times m$, such that:

$$\boldsymbol{y}(\boldsymbol{u}) = f(\boldsymbol{u}) = \boldsymbol{M}\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{y}$ is the n dimensional task outcome vector and $\boldsymbol{u}$ is the m dimensional motor command vector. We assume motor commands are generated from an n dimensional desired outcome, $\boldsymbol{y}^*$. The desired outcome is mapped to a motor command $\boldsymbol{u}$ with the inverse model function $\boldsymbol{g}$, represented by a $m \times n$ matrix $\boldsymbol{W}$, via the equation:

$$\boldsymbol{u}(\boldsymbol{y}^*) = g(\boldsymbol{y}^*) = \boldsymbol{W}\boldsymbol{y}^*. \tag{2}$$

The inverse model matrix $\boldsymbol{W}$ can be updated by applying gradient descent on the cost function $J_e = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e}$, where $\boldsymbol{e}$ is the error vector $\boldsymbol{e} = \boldsymbol{y}^* - \boldsymbol{y}$ between the desired and the actual outcome. The gradient of the cost function J_e in relation to $\boldsymbol{W}$ can be written as:

$$\begin{aligned}\nabla_{\boldsymbol{W}} J_e &= \left(\frac{\partial J_e}{\partial \boldsymbol{W}}\right)^T = \left(\frac{1}{2}\frac{\partial \boldsymbol{e}^T\boldsymbol{e}}{\partial \boldsymbol{W}}\right)^T = \left(\boldsymbol{e}^T\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{W}}\right)^T = \frac{\partial \boldsymbol{e}^T}{\partial \boldsymbol{W}}\boldsymbol{e} \\ &= -\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{W}}(\boldsymbol{y}^* - \boldsymbol{y}) = -\frac{\partial \boldsymbol{u}^T}{\partial \boldsymbol{W}}\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}(\boldsymbol{y}^* - \boldsymbol{y}) = \\ &\quad -\frac{\partial \boldsymbol{y}^{*T}\boldsymbol{W}^T}{\partial \boldsymbol{W}}\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}(\boldsymbol{y}^* - \boldsymbol{y}) = -\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}(\boldsymbol{y}^* - \boldsymbol{y})\boldsymbol{y}^{*T} \\ &= -\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}\boldsymbol{e}\boldsymbol{y}^{*T},\end{aligned} \tag{3}$$

so the update of the inverse model depends on the terms $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}\boldsymbol{e}$, which we call a motor correction term, also written as $\Delta\boldsymbol{u}$, and $\boldsymbol{y}^{*T}$, which is the input of the inverse model. The term $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}$ of the motor correction is called a sensitivity derivative (Abdelghani et al., 2008), which relates changes in the task space to the required changes in the motor command space. From Eq. (1), we know that the true sensitivity derivative is $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}} = \boldsymbol{M}^T$, meaning that the ideal motor correction is $\Delta\boldsymbol{u}^* = \boldsymbol{M}^T\boldsymbol{e}$ and the ideal gradient of the cost function is $\nabla_{\boldsymbol{W}} J_e = -\boldsymbol{M}^T\boldsymbol{e}\boldsymbol{y}^{*T}$, which is a $m \times n$ matrix with the same shape as the inverse model $\boldsymbol{W}$. Given the gradient of the cost function above, the update of the inverse model in one epoch of training can be written as:

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta_i\nabla_{\boldsymbol{W}} J_e = \boldsymbol{W}(k) + \eta_i\boldsymbol{M}^T\boldsymbol{e}\boldsymbol{y}^{*T}, \tag{4}$$

with η_i being a scalar learning rate for the inverse model.

We assume that the learning system does not have direct knowledge of the true sensitivity derivative $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}$. Instead, the learning system possesses a forward model, through which an estimate $\widehat{\boldsymbol{y}}$ of the task outcome given a motor command $\boldsymbol{u}$ can be obtained, via the $n \times m$ matrix $\widehat{\boldsymbol{M}}$:

$$\widehat{\boldsymbol{y}}(\boldsymbol{u}) = \widehat{\boldsymbol{M}}\boldsymbol{u}. \tag{5}$$

By backpropagating the task errors through the forward model, we can define that $\frac{\partial \widehat{\boldsymbol{y}}^T}{\partial \boldsymbol{u}} = \widehat{\boldsymbol{M}}^T$, and we can use this estimate of the sensitivity derivative instead of the true sensitivity derivative $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}$ to update the inverse model $\boldsymbol{W}$ and reduce the cost function. The sufficient condition for the estimated sensitivity derivative to reduce the cost function is that $\boldsymbol{e}^T\boldsymbol{M}\widehat{\boldsymbol{M}}^T\boldsymbol{e} > 0$, that is, the motor correction $\Delta\boldsymbol{u} = \widehat{\boldsymbol{M}}^T\boldsymbol{e} = \frac{\partial \widehat{\boldsymbol{y}}^T}{\partial \boldsymbol{u}}\boldsymbol{e}$ has an angle of less than 90° with the ideal motor correction $\Delta\boldsymbol{u}^* = \boldsymbol{M}^T\boldsymbol{e} = \frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{u}}\boldsymbol{e}$ (Lillicrap et al., 2016). This condition ensures that the update follows roughly the same direction of the gradient of the cost, therefore reducing the cost and the error of the inverse model.

We can decompose the motor correction into components belonging to the task-potent and task-null subspaces of motor commands. Decomposing the ideal motor correction $\Delta\boldsymbol{u}^*$, we obtain:

$$\Delta\boldsymbol{u}^* = \Delta\boldsymbol{u}^*_{\mathrm{TP}} + \Delta\boldsymbol{u}^*_{\mathrm{TN}} = \boldsymbol{M}^\dagger\boldsymbol{M}\Delta\boldsymbol{u}^* + \left(\boldsymbol{I} - \boldsymbol{M}^\dagger\boldsymbol{M}\right)\Delta\boldsymbol{u}^*, \tag{6}$$

where $\Delta\boldsymbol{u}^*_{\mathrm{TP}}$ is the task-potent component, $\Delta\boldsymbol{u}^*_{\mathrm{TN}}$ is the task-null component, and $\boldsymbol{M}^\dagger$ is the right pseudoinverse of $\boldsymbol{M}$, $\boldsymbol{M}^\dagger =$

$\boldsymbol{M}^T\left(\boldsymbol{M}\boldsymbol{M}^T\right)^{-1}$. We can show that, for the ideal motor correction $\Delta\boldsymbol{u}^* = \boldsymbol{M}^T\boldsymbol{e}$, the task-null component of the correction will be:

$$\begin{aligned}\Delta\boldsymbol{u}^*_{\mathrm{TN}} &= \left(\boldsymbol{I}-\boldsymbol{M}^\dagger\boldsymbol{M}\right)\boldsymbol{M}^T\boldsymbol{e} = \left(\boldsymbol{M}^T-\boldsymbol{M}^T\left(\boldsymbol{M}\boldsymbol{M}^T\right)^{-1}\boldsymbol{M}\boldsymbol{M}^T\right)\boldsymbol{e}\\ &= \left(\boldsymbol{M}^T-\boldsymbol{M}^T\right)\boldsymbol{e} = 0,\end{aligned} \tag{7}$$

that is, using the true sensitivity derivative in the motor correction will yield a motor correction fully in the task-potent space of motor commands. This can be further shown by calculating the task-potent component of the ideal motor correction:

$$\Delta\boldsymbol{u}^*_{\mathrm{TP}} = \boldsymbol{M}^\dagger\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{e} = \boldsymbol{M}^T\left(\boldsymbol{M}\boldsymbol{M}^T\right)^{-1}\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{e} = \boldsymbol{M}^T\boldsymbol{e}, \tag{8}$$

that is, the task-potent component of the ideal motor correction is the entire ideal motor correction. All changes on the inverse model using this ideal motor correction will affect the inverse model's projection in the task-potent space, with no change in its projection in the task-null space.

Likewise, the motor correction $\Delta\boldsymbol{u} = \widehat{\boldsymbol{M}}^T\boldsymbol{e}$ generated using the sensitivity derivative derived from the forward model can also be decomposed into its task-potent and task-null components, $\Delta\boldsymbol{u}_{\mathrm{TP}} = \boldsymbol{M}^\dagger\boldsymbol{M}\widehat{\boldsymbol{M}}^T\boldsymbol{e}$ and $\Delta\boldsymbol{u}_{\mathrm{TN}} = \left(\boldsymbol{I}-\boldsymbol{M}^\dagger\boldsymbol{M}\right)\widehat{\boldsymbol{M}}^T\boldsymbol{e}$, respectively. Because the ideal motor correction $\Delta\boldsymbol{u}^*$ is entirely in the task-potent space, and points in the direction of the gradient of the cost function, the task-potent component of the motor correction $\Delta\boldsymbol{u}$ will affect changes in the inverse model $\boldsymbol{W}$ that will cause changes in the task outcome, and in the cost. Its task-null component, however, is perpendicular to the gradient, so that its changes in the inverse model will not affect either the inverse model's task outcomes or cost.

2.2.2. Effect of amount of exploration noise on the training of the forward model

The forward model matrix $\widehat{\boldsymbol{M}}$ can be updated using gradient descent on the cost function $J_{\hat{\boldsymbol{e}}} = \frac{1}{2}\hat{\boldsymbol{e}}^T\hat{\boldsymbol{e}}$, where $\hat{\boldsymbol{e}}$ is the prediction error vector $\hat{\boldsymbol{e}} = \boldsymbol{y}-\widehat{\boldsymbol{y}}$ between the actual and predicted outcomes. The gradient of the cost function $J_{\hat{\boldsymbol{e}}}$ in relation to $\widehat{\boldsymbol{M}}$ can be written as:

$$\begin{aligned}\nabla_{\widehat{\boldsymbol{M}}}J_{\hat{\boldsymbol{e}}} &= \left(\frac{\partial J_{\hat{\boldsymbol{e}}}}{\partial\widehat{\boldsymbol{M}}}\right)^T = \left(\frac{1}{2}\frac{\partial\hat{\boldsymbol{e}}^T\hat{\boldsymbol{e}}}{\partial\widehat{\boldsymbol{M}}}\right)^T = \left(\hat{\boldsymbol{e}}^T\frac{\partial\hat{\boldsymbol{e}}}{\partial\widehat{\boldsymbol{M}}}\right)^T = \frac{\partial\hat{\boldsymbol{e}}^T}{\partial\widehat{\boldsymbol{M}}}\hat{\boldsymbol{e}}\\ &= -\frac{\partial\widehat{\boldsymbol{y}}^T}{\partial\widehat{\boldsymbol{M}}}\left(\boldsymbol{y}-\widehat{\boldsymbol{y}}\right) =\\ &\quad -\frac{\partial\boldsymbol{u}^T\widehat{\boldsymbol{M}}^T}{\partial\widehat{\boldsymbol{M}}}\left(\boldsymbol{y}-\widehat{\boldsymbol{y}}\right) = -\left(\boldsymbol{y}-\widehat{\boldsymbol{y}}\right)\boldsymbol{u}^T.\end{aligned} \tag{9}$$

Given a scalar learning rate η_{f}, the update of the forward model can be written as:

$$\begin{aligned}\widehat{\boldsymbol{M}}(k+1) &= \widehat{\boldsymbol{M}}(k) - \eta_{\mathrm{f}}\nabla_{\widehat{\boldsymbol{M}}}J_{\hat{\boldsymbol{e}}} = \widehat{\boldsymbol{M}}(k) + \eta_{\mathrm{f}}\left(\boldsymbol{y}(k)-\widehat{\boldsymbol{y}}(k)\right)\boldsymbol{u}(k)^T\\ &= \widehat{\boldsymbol{M}}(k) + \eta_{\mathrm{f}}\hat{\boldsymbol{e}}(k)\,\boldsymbol{u}(k)^T.\end{aligned} \tag{10}$$

We can observe from Eq. (10) that the update of the forward model depends on the prediction error $\hat{\boldsymbol{e}}(k)$ and on the motor command $\boldsymbol{u}(k)$. Thus, the update of the forward model depends on the active exploration of motor commands, generating task outcomes and sensory prediction errors, which are then used for learning. Sensory prediction errors are thought to be essential in the adaptation of internal models (Shadmehr et al., 2010), and it has been observed that the amount of motor exploration is correlated with motor learning (Dal'Bello & Izawa, 2021; Singh et al., 2016; Wu et al., 2014).

Instead of using a single motor command to update the forward model at every learning epoch, we can update it by using multiple samples of motor commands at every epoch. Defining S as the number of motor command samples to use in the update, we can obtain the following equation for the update of the forward model:

$$\begin{aligned}\widehat{\boldsymbol{M}}(k+1) &= \widehat{\boldsymbol{M}}(k) + \frac{\eta_{\mathrm{f}}}{S}\sum_{s=1}^{S}\hat{\boldsymbol{e}}_s(k)\,\boldsymbol{u}_s(k)^T\\ &= \widehat{\boldsymbol{M}}(k) + \frac{\eta_{\mathrm{f}}}{S}\sum_{s=1}^{S}\left(\boldsymbol{M}\boldsymbol{u}_s(k) - \widehat{\boldsymbol{M}}(k)\,\boldsymbol{u}_s(k)\right)\boldsymbol{u}_s(k)^T\\ &= \widehat{\boldsymbol{M}}(k)\left(I - \frac{\eta_{\mathrm{f}}}{S}\sum_{s=1}^{S}\boldsymbol{Z}_s(k)\right) + \boldsymbol{M}\frac{\eta_{\mathrm{f}}}{S}\sum_{s=1}^{S}\boldsymbol{Z}_s(k),\end{aligned} \tag{11}$$

where $\hat{\boldsymbol{e}}_s(k)$ is the sth prediction error, $\boldsymbol{u}_s(k)$ is the sth motor command at learning epoch k, and $\boldsymbol{Z}_s(k) = \boldsymbol{u}_s(k)\,\boldsymbol{u}_s(k)^T$. Assuming that the motor commands $\boldsymbol{u}_s(k)$ used in the update of the forward model are independent and identically distributed random variables, having the same probability distribution at every sample s and every epoch k, the term $\frac{1}{S}\sum_{s=1}^{S}\boldsymbol{Z}_s(k)$ can be replaced by the expected value $E[\boldsymbol{Z}_s(k)] = \overline{\boldsymbol{Z}}$ when the number of samples S is large. On this basis, Eq. (11) can be rewritten as:

$$\widehat{\boldsymbol{M}}(k+1) = \widehat{\boldsymbol{M}}(k)\left(I-\eta_{\mathrm{f}}\overline{\boldsymbol{Z}}\right) + \eta_{\mathrm{f}}\boldsymbol{M}\overline{\boldsymbol{Z}}. \tag{12}$$

2.2.3. Effect of a limitation of the neural representation of the motor correction on the update of the inverse model

We will now analyze the effects of our assumption that, because of the encoding of the motor correction in the brain via the firing rate of a population of neurons (Dayan & Abbott, 2001; Inoue et al., 2016), biophysical constraints of neurons, such as a limit in their firing rates, can limit the amplitude of an encoded motor correction signal. Given our motor correction $\Delta\boldsymbol{u} = \widehat{\boldsymbol{M}}^T\boldsymbol{e}$ generated using the sensitivity derivative derived from the forward model, this sensitivity derivative can be decomposed on its task-potent and task-null components using Eq. (6), resulting in:

$$\Delta\boldsymbol{u} = \Delta\boldsymbol{u}_{\mathrm{TP}} + \Delta\boldsymbol{u}_{\mathrm{TN}} = \boldsymbol{M}^\dagger\boldsymbol{M}\widehat{\boldsymbol{M}}^T\boldsymbol{e} + \left(\boldsymbol{I}-\boldsymbol{M}^\dagger\boldsymbol{M}\right)\widehat{\boldsymbol{M}}^T\boldsymbol{e}. \tag{13}$$

Let us now assume that there exists a maximum allowed norm a for the motor correction, so that when this norm is surpassed, we change the gain of the motor correction without changing its direction. The neural representation of the norm-limited motor correction will then be:

$$\widehat{\Delta\boldsymbol{u}} = \frac{a}{\|\Delta\boldsymbol{u}\|}\Delta\boldsymbol{u}, \tag{14}$$

where the gain $\frac{a}{\|\Delta\boldsymbol{u}\|}$ limits the norm of the motor correction to a, and the norm of the motor correction signal is given by:

$$\|\Delta\boldsymbol{u}\| = \sqrt{\|\Delta\boldsymbol{u}_{\mathrm{TP}}\|^2 + \|\Delta\boldsymbol{u}_{\mathrm{TN}}\|^2}. \tag{15}$$

The update of the inverse model in this epoch can be written as:

$$\Delta\boldsymbol{W}(k) = \boldsymbol{W}(k+1) - \boldsymbol{W}(k) = \eta_{\boldsymbol{W}}\widehat{\Delta\boldsymbol{u}}\boldsymbol{y}^{*T}. \tag{16}$$

This limitation of the norm of the motor correction shares similarities with the technique of gradient clipping proposed in the field of machine learning (Pascanu et al., 2013). However, gradient clipping is traditionally applied to a single neural network, and with the objective of avoiding the problem of exploding gradients, while in our approach we apply it to the backpropagation of the motor correction between two neural networks, from the forward model to the inverse model, with the objective of mimicking biophysical constraints of the encoding of the motor correction signal in the human brain.

The task-potent component of the update of the inverse model, which will effectively reduce the cost of the inverse model, will

be:

$$\Delta \boldsymbol{W}_{\mathrm{TP}}(k) = \eta_{\boldsymbol{W}} \widehat{\Delta \boldsymbol{u}}_{\mathrm{TP}} \boldsymbol{y}^{*T} = \eta_{\boldsymbol{W}} \frac{a}{\|\Delta \boldsymbol{u}\|} \Delta \boldsymbol{u}_{\mathrm{TP}} \boldsymbol{y}^{*T} = \frac{a}{\sqrt{\|\Delta \boldsymbol{u}_{\mathrm{TP}}\|^2 + \|\Delta \boldsymbol{u}_{\mathrm{TN}}\|^2}} \Delta \boldsymbol{u}_{\mathrm{TP}} \boldsymbol{y}^{*T}. \tag{17}$$

For the update to cause a larger change in the inverse model, reducing the cost more in a single epoch, it is desirable for the task-potent component of the update (parallel to the gradient of the cost function) to be as large as possible. For fixed $\Delta \boldsymbol{u}_{\mathrm{TP}}$ and $\boldsymbol{y}^{*T}$ in Eq. (17), this can be achieved by minimizing $\|\Delta \boldsymbol{u}_{\mathrm{TN}}\|^2$, that is, minimizing the norm of the task-null component of the motor correction.

2.2.4. Updating the forward model with an exploration of the entire motor command space guarantees efficient motor correction of inverse model

We can minimize the norm of the task-null component of the motor correction $\Delta \boldsymbol{u}$ by updating the forward model, using Eq. (12). Let us compare the motor correction $\Delta \boldsymbol{u}$ before and after the update of the forward model, supposing a fixed inverse model error $\boldsymbol{e}$, and using the fact that the matrix $\overline{\boldsymbol{Z}}$ is symmetrical:

$$\Delta \boldsymbol{u}(k+1) = \widehat{\boldsymbol{M}}(k+1)^T \boldsymbol{e} = \left(\widehat{\boldsymbol{M}}(k)\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)\right)^T \boldsymbol{e} + \eta_{\mathrm{f}} \left(\boldsymbol{M} \overline{\boldsymbol{Z}}\right)^T \boldsymbol{e} = \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right) \widehat{\boldsymbol{M}}(k)^T \boldsymbol{e} + \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} \boldsymbol{M}^T \boldsymbol{e} = \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right) \Delta \boldsymbol{u}(k) + \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} \Delta \boldsymbol{u}^*, \tag{18}$$

so the updated motor correction $\Delta \boldsymbol{u}(k+1)$ can be defined in terms of the previous motor correction $\Delta \boldsymbol{u}(k)$ and the ideal motor correction $\Delta \boldsymbol{u}^*$.

Eq. (18) can be rewritten in terms of the initial motor correction at $k = 0$ as:

$$\Delta \boldsymbol{u}(k) = \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k \Delta \boldsymbol{u}(0) + \sum_{j=1}^{k} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^{j-1} \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} \Delta \boldsymbol{u}^*. \tag{19}$$

The second term in the equation, multiplying the ideal motor correction $\Delta \boldsymbol{u}^*$, can be rewritten in a form other than a sum. Defining $S_k = \sum_{j=1}^{k} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^{j-1} \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}$, we have:

$$\begin{aligned} S_k - \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right) S_k &= \sum_{j=1}^{k} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^{j-1} \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} - \sum_{j=1}^{k} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^{j} \\ \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} &= \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} - \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} = \eta_{\mathrm{f}} \overline{\boldsymbol{Z}} S_k \end{aligned} \tag{20}$$

and then, assuming that $\overline{\boldsymbol{Z}}$ has an inverse:

$$S_k = I - \overline{\boldsymbol{Z}}^{-1} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k \overline{\boldsymbol{Z}}, \tag{21}$$

so that Eq. (19) can be rewritten as:

$$\Delta \boldsymbol{u}(k) = \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k \Delta \boldsymbol{u}(0) + \left(I - \overline{\boldsymbol{Z}}^{-1} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k \overline{\boldsymbol{Z}}\right) \Delta \boldsymbol{u}^*. \tag{22}$$

The convergence of the term $\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k$, with k going to infinity, in Eq. (22) will depend on the eigenvalues of the matrix $I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}$ (de Boor, 2002), which can be represented by $\lambda\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right) = 1 - \eta_{\mathrm{f}} \lambda\left(\overline{\boldsymbol{Z}}\right)$. Given a symmetric matrix A, A^k converges to a finite matrix, with k going to infinity, if $|\lambda(A)| \leq 1$ (that is, all of its eigenvalues have a norm that is less than or equal to 1), with convergence to the zero matrix if $|\lambda(A)| < 1$ (that is, all of its eigenvalues have a norm strictly less than 1).

Let us now assume that the motor commands $\boldsymbol{u}_s(k)$ used in the update of the forward model are defined as:

$$\boldsymbol{u}_s(k) = \boldsymbol{b} + \boldsymbol{n}_s(k), \boldsymbol{n}_s(k) \sim N(\boldsymbol{0}, \Sigma), \tag{23}$$

where $\boldsymbol{b}$ is a constant bias vector and $\boldsymbol{n}_s(k)$ is exploration noise, represented by a random variable from a multivariate normal distribution with a mean vector of 0 and covariance matrix Σ. Given this,

$$\overline{\boldsymbol{Z}} = E\left[\boldsymbol{u}_s(k) \boldsymbol{u}_s(k)^T\right] = \Sigma + \boldsymbol{b}\boldsymbol{b}^T. \tag{24}$$

Being composed of a sum of two positive semidefinite matrices, $\overline{\boldsymbol{Z}}$ is then also positive semidefinite, with the property that $\lambda\left(\overline{\boldsymbol{Z}}\right) \geq 0$, that is, all of its eigenvalues are nonnegative. If we have any $\lambda\left(\overline{\boldsymbol{Z}}\right) = 0$, this would give us at least one $\lambda\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right) = 1$, which would make $\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k$ fail to converge to the zero matrix (in this case, the inverse of $\overline{\boldsymbol{Z}}$ does not exist, so we cannot rewrite Eq. (19) as Eq. (22)). A way to guarantee that $\lambda\left(\overline{\boldsymbol{Z}}\right) > 0$ (i.e., all eigenvalues of $\overline{\boldsymbol{Z}}$ are strictly positive) is to use motor commands $\boldsymbol{u}_s(k)$ that span the entire motor command space, which is equivalent to making the matrix $\overline{\boldsymbol{Z}}$ full rank. This can be achieved by having non-zero variance in all dimensions of the motor command covariance matrix Σ, even with the presence of correlation in the noise (although not a full correlation, because this would constrain the dimensions in which the noise is generated). This enables us to write the following:

The entire motor command space explored $\Rightarrow$ $\overline{\boldsymbol{Z}}$ *full rank*

$$\Rightarrow \lambda\left(\overline{\boldsymbol{Z}}\right) > 0 \Rightarrow \lim_{k \to \infty} \left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k = 0, 0 < \eta_{\mathrm{f}} < \min\left(\frac{2}{\lambda\left(\overline{\boldsymbol{Z}}\right)}\right), \tag{25}$$

where $\boldsymbol{0}$ is a $m \times m$ matrix of zeros. Thus, when the entire motor command space is explored, $\overline{\boldsymbol{Z}}$ will be full rank, and we will have:

$$\lim_{k \to \infty} \Delta \boldsymbol{u}(k) = 0 \Delta \boldsymbol{u}(0) + \left(I - \overline{\boldsymbol{Z}}^{-1} \boldsymbol{0} \overline{\boldsymbol{Z}}\right) \Delta \boldsymbol{u}^* = \Delta \boldsymbol{u}^*, \tag{26}$$

that is, the forward model motor correction converges to the ideal motor correction $\Delta \boldsymbol{u}^*$, thus guaranteeing that its use in the update of the inverse model $\boldsymbol{W}$ will reduce its error and cost.

If the condition of Eq. (25), that $\overline{\boldsymbol{Z}}$ is full rank, is not met, then $\overline{\boldsymbol{Z}}$ will have at least one eigenvalue equal to 0, which would make $I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}$ have at least one eigenvalue equal to 1. Although this is sufficient to make $\left(I - \eta_{\mathrm{f}} \overline{\boldsymbol{Z}}\right)^k$ converge to a finite matrix with k going to infinity (de Boor, 2002), this finite matrix will not be the zero matrix, so the convergence of $\Delta \boldsymbol{u}(k)$ to the ideal motor correction $\Delta \boldsymbol{u}^*$ is not guaranteed. In particular, without any exploration of motor commands, such as when the motor commands $\boldsymbol{u}_s(k)$ are just composed of the bias vector $\boldsymbol{b}$ in Eq. (23), then the rank of $\overline{\boldsymbol{Z}} = \boldsymbol{b}\boldsymbol{b}^T$ will be 1, which is far from the requirement that $\overline{\boldsymbol{Z}}$ is full rank (assuming the dimension of the motor command space $m > 1$). Therefore, motor exploration is essential in the training of a forward model, with exploration of the entire motor command space guaranteeing that the sensitivity derivative obtained from the forward model will lead to the decrease of the error and cost of the inverse model.

2.2.5. Not exploring the task-null space of motor commands during update of the forward model does not guarantee efficient motor correction of inverse model

In the previous section, we showed that exploring the entire motor command space leads to the forward model's motor correction converging to the ideal motor correction, which does not have a task-null component (Eq. (7)). However, when the task-null space of motor commands is not explored, the convergence of the task-null component of the motor correction to zero is not guaranteed. This derives directly from the equation of the update of the forward model, Eq. (11). Assuming a constant inverse model error $\boldsymbol{e}$, we have:

$$\Delta \boldsymbol{u}(k+1) = \widehat{\boldsymbol{M}}(k+1)^T \boldsymbol{e} = \widehat{\boldsymbol{M}}(k)^T \boldsymbol{e} + \frac{\eta_{\mathrm{f}}}{S} \left(\sum_{s=1}^{S} \hat{\boldsymbol{e}}_s(k) \boldsymbol{u}_s(k)^T\right)^T \boldsymbol{e} = \Delta \boldsymbol{u}(k) + \frac{\eta_{\mathrm{f}}}{S} \sum_{s=1}^{S} \boldsymbol{u}_s(k) \hat{\boldsymbol{e}}_s(k)^T \boldsymbol{e}. \tag{27}$$

Assuming that the task-null space is not explored, we can write $\boldsymbol{u}_s(k) = \boldsymbol{M}^T \boldsymbol{a}(k)$, with $\boldsymbol{a}(k)$ being a $n \times 1$ vector; that is, $\boldsymbol{u}_s(k)$ is a linear combination of the motor commands belonging to the task-potent space. Given this, and using Eq. (13), the task-null component of the updated motor correction will be:

$$\begin{aligned}\Delta\boldsymbol{u}_{\mathrm{TN}}(k+1) &= \left(I - \boldsymbol{M}^{\dagger}\boldsymbol{M}\right)\Delta\boldsymbol{u}(k+1) = \left(I - \boldsymbol{M}^{\dagger}\boldsymbol{M}\right)\Delta\boldsymbol{u}(k) \\ &\quad + \frac{\eta_{\mathrm{f}}}{S}\left(I - \boldsymbol{M}^{\dagger}\boldsymbol{M}\right)\sum_{s=1}^{S}\boldsymbol{u}_s(k)\,\hat{\boldsymbol{e}}_s(k)^T\boldsymbol{e} \\ &= \Delta\boldsymbol{u}_{\mathrm{TN}}(k) + \frac{\eta_{\mathrm{f}}}{S}\sum_{s=1}^{S}\left(\boldsymbol{M}^T - \boldsymbol{M}^T\right)\boldsymbol{a}(k)\,\hat{\boldsymbol{e}}_s(k)^T\boldsymbol{e} \\ &= \Delta\boldsymbol{u}_{\mathrm{TN}}(k),\end{aligned} \tag{28}$$

that is, motor commands with a zero task-null component will never generate a change in the task-null component of $\Delta\boldsymbol{u}$. When the task-null space is not explored, the only way $\Delta\boldsymbol{u}$ will have no task-null component is if $\Delta\boldsymbol{u}_{\mathrm{TN}}(0) = \boldsymbol{0}$ (i.e., if the initial motor correction $\Delta\boldsymbol{u}(0)$ already has a zero task-null component). From this, we can conclude that exploration of task-null motor commands is necessary if the aim is to reduce the task-null component of the motor correction to zero, enabling an efficient update of the inverse model when the norm of the motor correction is limited.

2.2.6. Predictions of our framework

In summary, our theoretical analysis enabled us to make the following predictions regarding the usage of motor exploration on learning:

- Prediction 1: the task-potent component of the motor correction signal used to train the inverse model is responsible for reducing the test cost of the inverse model;
- Prediction 2: when the motor correction used to train the inverse model has a limited norm, a motor correction with a smaller task-null component leads to a faster decrease of the cost of the inverse model;
- Prediction 3: a larger motor command used in the training of a forward model/sensitivity derivative will lead to a larger update, and to a faster decrease of its test cost;
- Prediction 4: when the entire motor command space is explored in the training of a forward model/sensitivity derivative, its motor correction will converge to the ideal motor correction, with no task-null component;
- Prediction 5: if the task-null space of motor commands is not explored during the training of a forward model/sensitivity derivative, the task-null component of its motor correction will never be minimized.

2.2.7. Simulations

To show some of the features of the learning system discussed above, we performed simulations of a simple linear motor task in which the motor command space is 2D and the task outcome (and desired outcome) space is 1D. The environment used in the task (Eq. (1)) was arbitrarily defined as $\boldsymbol{M} = \begin{bmatrix}0 & 1\end{bmatrix}$, so that the Y axis of the motor commands space is the task-potent space and the X axis is the task-null space of motor commands. We decided to focus our simulations on the problem of de novo learning, where the initial sensitivity derivative, here represented by the forward model, does not generate an adequate motor correction that will lead to a decrease in the inverse model's test cost. To that end, we initialized our forward model at $\widehat{\boldsymbol{M}} = \begin{bmatrix}1 & 0\end{bmatrix}$ (Eq. (5)), that is, belonging only to the task-null space. The inverse model is initialized at $\boldsymbol{W} = \begin{bmatrix}0.5 & 0\end{bmatrix}^T$ (Eq. (2)). From a de novo learning point of view, the original task-potent space is the X axis (the new task-null space), and the original task-null space is the Y axis (the new task-potent space).

We conducted two separate simulations of this simple linear motor task. In simulation 1 (Fig. 4), we aimed to verify some of the predictions regarding the training of the forward model shown earlier in our equations by analyzing the training of the forward model under different combinations of exploration noise in the task-potent and task-null spaces. In a single epoch, we first updated the forward model using a batch of $S = 10$ motor command samples, with a learning rate $\eta_{\mathrm{f}} = 0.05$, and the inverse model was then updated, in the same epoch, with the desired outcome of 0.5 and with a learning rate $\eta_{\mathrm{i}} = 0.2$. Motor commands used in the training of the forward model were obtained from a bivariate normal distribution with a mean vector of 0 and variances in the X and Y axes (task-null and task-potent axes respectively) being combinations among the possible values of 0, 0.1, 0.2, 0.3, and 0.4. The networks were trained for a total of 250 epochs.

In simulation 2 (Fig. 5), we aimed to replicate a scenario which is possibly closer to how motor commands would be generated in a biological system. Again, both forward and inverse models are updated in the same epoch. However, in this simulation the update of the forward model occurs with a single motor command. The motor command used in the update of the forward model is obtained by first using the state of the inverse model at the current epoch, with the desired outcome of 0.5, to generate a motor command, to which noise is added, with the same possible combinations of variances in the task-potent and task-null spaces as in the previous simulations. This noisy motor command is then used in the update of the forward model. In addition, the noisy motor command is also used in the training of the inverse model: this motor command is used to obtain the (noisy) task outcome, which is used to calculate the task error used in the update. The learning rates were the same as those used in the previous simulations.

Each simulation is repeated 20 times for each combination of exploration noise variance in the task-potent and task-null spaces. We calculate the average forward and inverse models across the 20 repetitions for every epoch of the simulation, to analyze the models' convergence. At every epoch we also calculated the test cost of the forward model, using 50 motor commands obtained from a bivariate normal distribution with a mean vector of 0 and a variance of 0.5 on both axes. The test cost of the inverse model was calculated with the four desired outcomes of -0.75, -0.25, 0.25 and 0.75.

2.3. Arm reaching task

2.3.1. Task summary

To examine whether the recently reported effect of exploration on motor learning might be explained by our theoretical outcomes, we simulated three different motor tasks under our proposed framework. The first motor task we simulated was the typical visuomotor learning task described by Singh et al. (2016) (Fig. 6). In this experiment, participants made reaching movements with their arms to eight targets shown on a screen, with their real arms obscured. During the experiment, the participants' arm joint angles were recorded (Fig. 6A), as well as their hand position. The experiment was divided into three phases (Fig. 6B): a preadaptation (or baseline) phase, in which the positions of the real hand and the cursor shown on the screen were the same; a visuomotor adaptation phase, in which a visuomotor rotation of 45° was applied to the cursor; and a postadaptation (washout) phase, in which the visuomotor rotation was removed. During the task, the main performance metric measured was the error at peak velocity (Fig. 6C), defined as the perpendicular distance

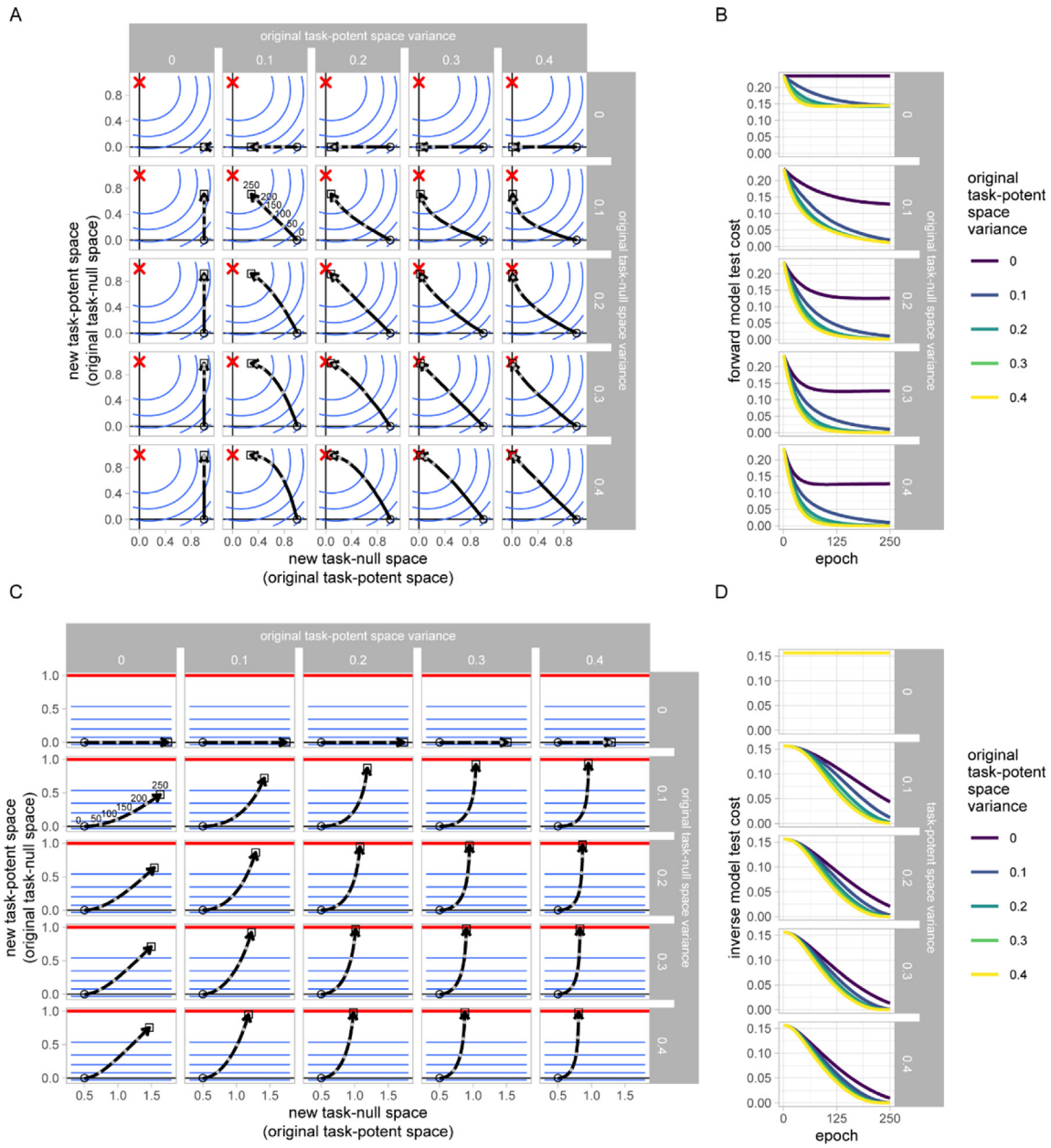

Fig. 4. Diagrams and results from simulation 1 of a simple linear motor task. In this simulation, both forward and inverse models are trained simultaneously, with the forward model trained with a batch of 10 random motor commands at every epoch. (A) Update of the forward model through learning, in all combinations of variances of exploration noise in the task-potent and task-null spaces. Blue lines represent contours of the test cost across the possible configurations of the forward model, with the red cross indicating the configuration with minimum cost (the ideal forward model, equal to the task environment). Empty circles represent the initial configuration of the forward model, and empty squares represent the final configuration. Gray circles represent intermediary configurations at every 50 epochs. (B) The test cost of the forward models across all combinations of exploration noise in the task-potent and task-null spaces. (C) Update of the inverse model through learning, in all combinations of variances of exploration noise in the task-potent and task-null spaces (note that exploration noise was only used during the training of the forward model). The inverse model configuration with minimum cost is the red line. (D) Test cost of the inverse models across all combinations of exploration noise in the task-potent and task-null spaces. In all panels, lines are averages across 20 repetitions of each simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the hand trajectory at peak velocity from the straight line connecting the start position and the target position. Learning during the visuomotor adaptation phase was analyzed for each participant by fitting an exponential function to the error at peak velocity, with the equation:

$$f(n) = ae^{-\beta n}, \tag{29}$$

where n is the trial number, a is a scaling parameter and β is an estimated learning rate, with a larger estimated learning rate corresponding to a faster decrease in errors during the visuomotor adaptation phase.

The authors analyzed the motor variability in both task and null spaces of movements during the baseline phase. Variability in the task space was measured with the standard deviation of the error at peak velocity, defined above, across all trials in the baseline phase. For the null space, the authors first defined a 2D forward kinematics model in which the hand position $\boldsymbol{y}$ in the 2D task space is defined by the arm joint angles and segment lengths, and was calculated as follows:

$$\boldsymbol{y} = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos\theta_2 + l_3 \cos\theta_3 + l_4 \cos\theta_4 \\ l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_3 \sin\theta_3 + l_4 \sin\theta_4 \end{bmatrix}, \tag{30}$$

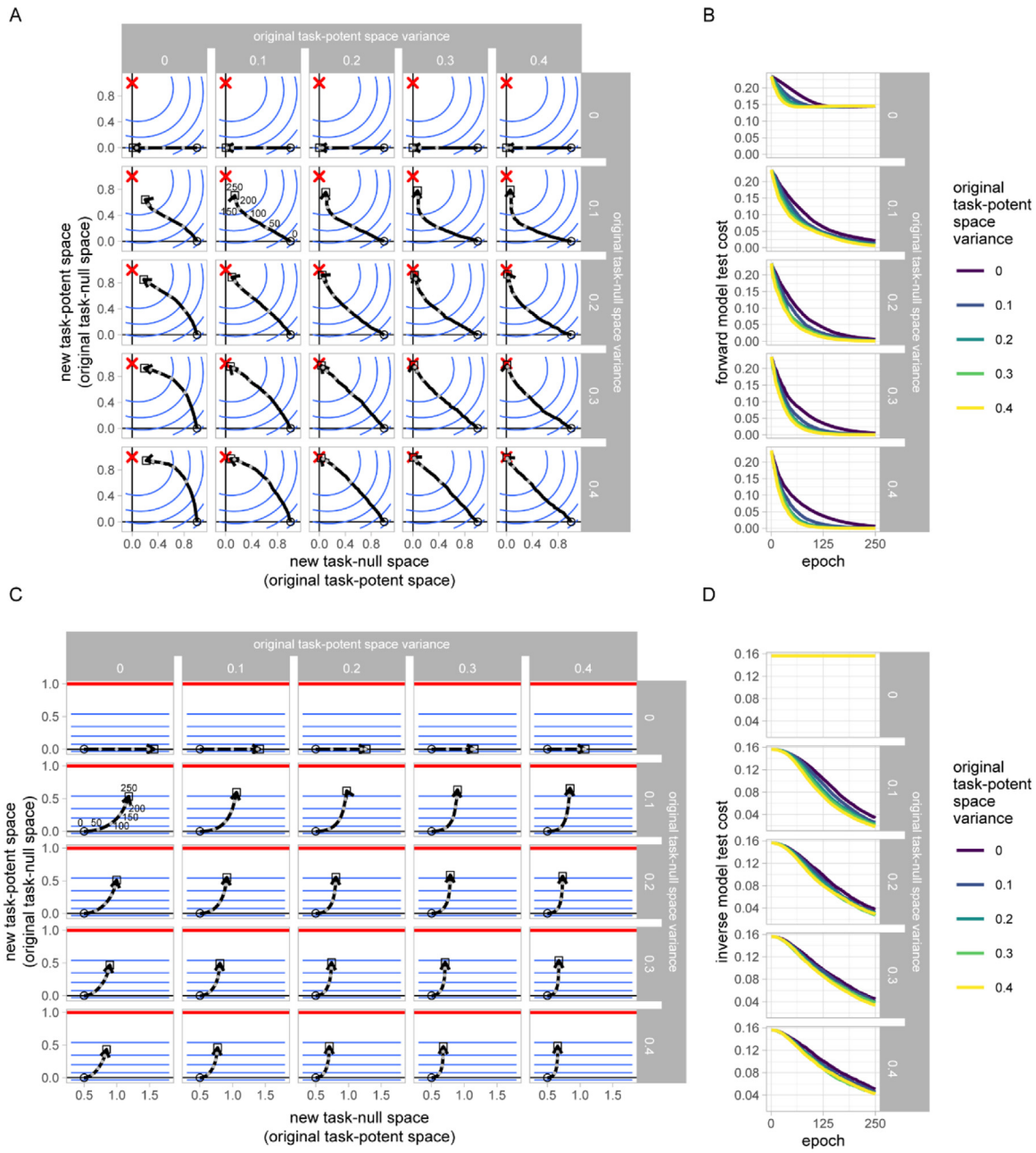

Fig. 5. Diagrams and results from simulation 2 of a simple linear motor task. In this simulation, both forward and inverse models are trained simultaneously. However, the forward model is updated with a single motor command, originating from the inverse model, with the addition of exploration noise. The noisy motor command is then also used to update the inverse model. (A) Update of the forward model through learning, in all combinations of variances of exploration noise in the task-potent and task-null spaces. Blue lines represent contours of the test cost across the possible configurations of the forward model, with the red cross indicating the configuration with minimum cost (the ideal forward model, equal to the task environment). Empty circles represent the initial configuration of the forward model, and empty squares represent the final configuration. Gray circles represent intermediary configurations at every 50 epochs. (B) Test cost of the forward models across all combinations of exploration noise in the task-potent and task-null spaces. (C) Update of the inverse model through learning, in all combinations of variances of exploration noise in the task-potent and task-null spaces (in this simulation, the motor command with added exploration noise was also used in the training of the inverse model). The inverse model configuration with minimum cost is the red line. (D) Test cost of the inverse models across all combinations of exploration noise in the task-potent and task-null spaces. In all panels, lines are averages across 20 repetitions of each simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where the pairs (l_j, θ_j) correspond to the jth arm segment length and joint angle, respectively, among the four defined arm segments/joints. The authors then calculated the mean arm joint configuration at peak velocity, $\overline{\boldsymbol{\theta}}^V$, across all baseline trials, and then computed the Jacobian matrix of the forward kinematics at the peak velocity, $J\left(\overline{\boldsymbol{\theta}}^V\right)$. Variability in the null space was then calculated by averaging the projection of the deviation from mean joint configuration, $\Delta\boldsymbol{\theta}_k = \overline{\boldsymbol{\theta}}^V - \boldsymbol{\theta}_k^V$ (where $\boldsymbol{\theta}_k^V$ is the joint configuration at trial k), on the null space of the Jacobian $J\left(\overline{\boldsymbol{\theta}}^V\right)$ across all baseline trials.

The main finding of Singh et al.'s (2016) experiment was that participants' null space variability exhibited a significant correlation with the estimated learning rate during the visuomotor adaptation (Fig. 6D), whereas task space variability did

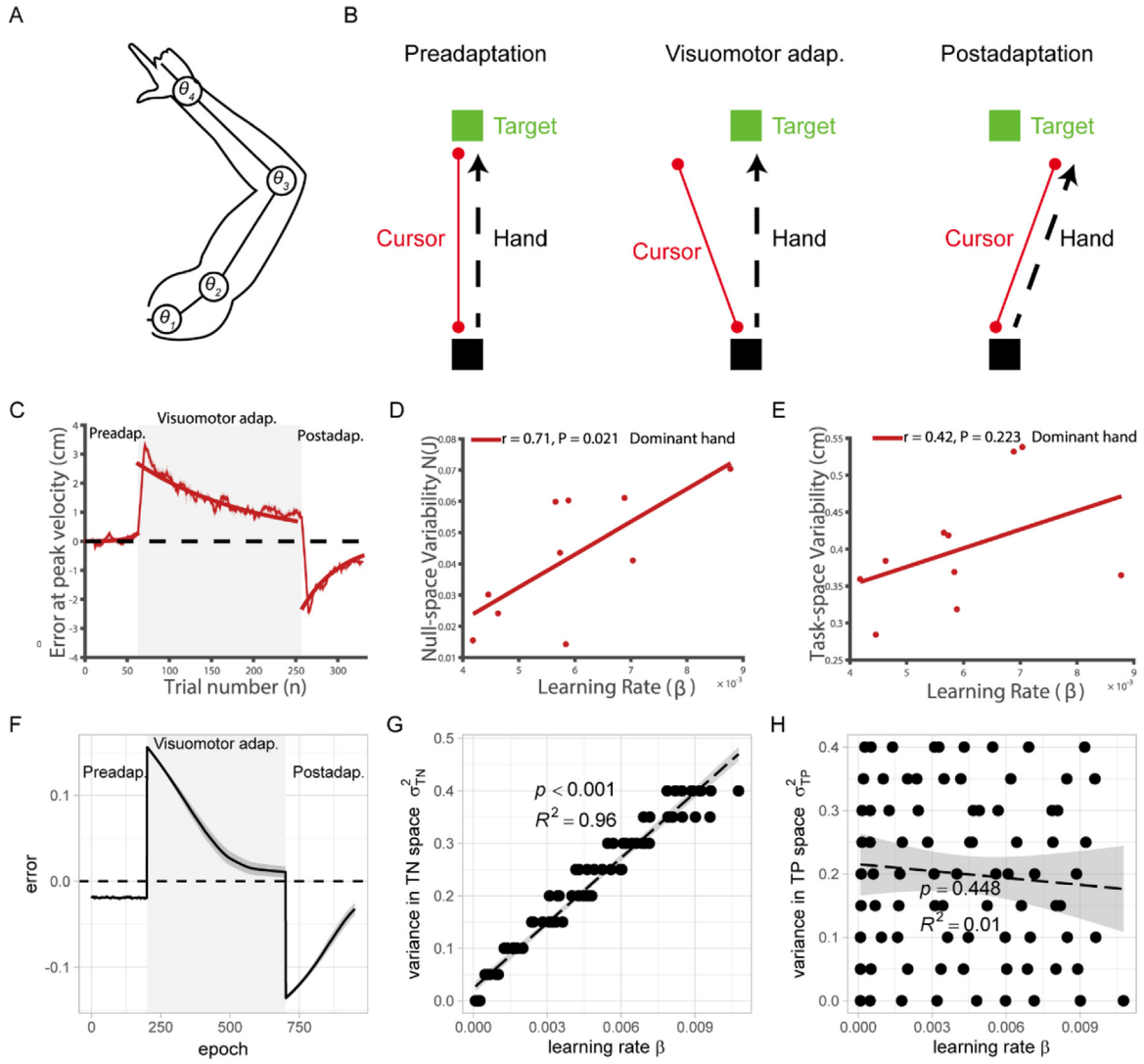

Fig. 6. Diagrams and results of the task from Singh et al. (2016). (A) The task involved participants making reaching movements to targets shown on a screen while their real arms were hidden. In addition to the position of their hands, the joint angles of the arm were also measured, from which they could estimate task-potent and task-null variabilities at the instant of peak hand velocity in the beginning of the experiment (during the preadaptation stage). (B) Their experiment was divided into three stages: preadaptation, where the positions of the hand and the displayed cursor are the same; visuomotor adaptation, where a constant visuomotor rotation is added to the cursor; and postadaptation, where again the positions of the hand and the cursor are the same. (C) Data from the eight-trial running mean error at peak velocity across participants. (D–E) Correlation between the estimated learning rate, estimated by fitting an exponential curve to the error at peak velocity during the visuomotor adaptation for each participant, and the variability in the null (D) and task (E) spaces. (F) Mean inverse model error across our simulations of the motor task The graph shows the mean error throughout the training of the inverse model across 20 simulations, where motor noise with a variance of 0.2 in both task-potent and task-null spaces was used to update the forward model before the training of the inverse model. The shaded region corresponds to one standard error. (G–H) Correlation between the estimated learning rate (estimated with the same method as the original experiment) and the variances in the task-null (G) and task-potent (H) spaces used in the motor noise during the training of the forward models in our simulations. Each point is an average across 20 simulations. Panels C, D, E were reproduced from Singh et al. (2016). TP: task-potent. TN: task-null.

not (Fig. 6E). Similar results were obtained in further force field adaptation experiments, with both dominant and nondominant hands (although the correlation in the nondominant hand was not significant). In addition, it was found that the nondominant hand, which had significantly less null space variability than the dominant hand (but not significantly different task space variability), also exhibited a significantly slower estimated learning rate during force field adaptation. The authors concluded that variability in the null space aided learning, and suggested that such null space variability may reflect active control and not only the biomechanical characteristics of the arm.

A possible interpretation of Singh et al.'s (2016) findings, under our framework, is that the participants' differences in null space variability reflected their certainty about which motor commands were task-null, with more certainty resulting in more variability allocated to the null space (Sternad, 2018). This certainty is also reflected in their internal sensitivity derivatives' task-null component, with more certainty leading to a smaller component in the task-null space. From our Prediction 2, we hypothesized that participants who were more certain about task-null movements (and with more task-null variability) would exhibit a sensitivity derivative with a smaller task-null component, and therefore the adaptation of their inverse models during the visuomotor adaptation phase would be faster, resulting in a positive correlation between the estimated learning rate and task-null variability.

2.3.2. Simulations

To simulate this task using our framework, we first defined the nonlinear mapping between joint angles and hand position

in a 2D plane. To achieve this, we used Eq. (30), with lengths of 0.1, 0.33, 0.27, and 0.08 for the four segments in our arm model. Next, we initialized an inverse model as an artificial neural network with three layers: an initial linear layer, whose input is the 2D desired outcome vector; a middle layer with eight neural units, using the hyperbolic tangent function, enabling the nonlinearities of the task environment to be captured; and a linear output layer, outputting the 4D motor command (in this case, the joint angles used in the arm) that will generate the outcome. To mimic the results of the experiment, in which task performance was good at the baseline phase (Fig. 6C), we initially trained the inverse model for 1000 epochs, with a learning rate of 0.1, with a 3×3 grid of desired outcomes, with the values among $(-0.2, 0.0, +0.2)$ in both the X and Y axes of the task space. The sensitivity derivative used in this initial training was the actual Jacobian of the environment calculated at the motor command generated by the inverse model. To evaluate the training of the inverse model, we defined a single target at $\boldsymbol{y}_{\text{tgt}} = \begin{bmatrix} 0 & 0.3 \end{bmatrix}^T$ and an initial arm position at $\boldsymbol{y}_0 = \begin{bmatrix} 0 & 0.1 \end{bmatrix}^T$, and analyzed the error as the perpendicular distance between the outcome of the inverse model using the target $\boldsymbol{y}_{\text{tgt}}$ as an input (the desired outcome) and the straight line connecting the starting position $\boldsymbol{y}_0$ and the target position $\boldsymbol{y}_{\text{tgt}}$, in a similar way to that performed in Singh et al.'s (2016) experiment. For simplicity, we evaluated this error at a single target, and our simulation did not include the arm dynamics to generate the entire arm trajectory during reaching, so we evaluated the error at the endpoint instead. The last 250 epochs of such initial training of the inverse model are shown in the baseline/preadaptation phase in Fig. 6F.

Next, we initialized our forward models to be used as sensitivity derivatives during the update of the inverse model at the visuomotor adaptation and postadaptation phases of training. We decided to represent the forward models of our simulations in a linearized way, approximating the Jacobian of the environment at the motor command $\bar{\boldsymbol{\theta}}$ obtained by inputting the previously defined target $\boldsymbol{y}_{\text{tgt}}$ in the inverse model trained for 1000 epochs. To initialize the forward models, we first calculated the Jacobian of the environment at the motor command $\bar{\boldsymbol{\theta}}$, $J\left(\bar{\boldsymbol{\theta}}\right)$, a 2×4 matrix. We then calculated a matrix representing the null space of the Jacobian, $J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)$, and 2×4 (the rows of this matrix represent changes in the default motor command $\bar{\boldsymbol{\theta}}$ which will not have an effect on the outcome), using singular value decomposition. We then initialized our forward models as the sum of the Jacobian $J\left(\bar{\boldsymbol{\theta}}\right)$ and the null space of the Jacobian $J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)$, so that forward models are initialized with a large projection in both task-potent and task-null spaces. Regarding our previous findings about the effect of a task-null component of the forward model in the update of the inverse model, in these simulations, it is desirable for the task-null component of the forward models to be reduced, and for the task-potent component to be maintained as it is.

For the training of the forward models, we used combinations of exploration noise variance in both task-potent and task-null spaces, among the nine possible values going from 0.0 to 0.4 with steps of 0.05, giving a total of 81 different combinations, with 20 repetitions of simulations for every combination of noise variance. To generate motor commands for the training of the forward models, we first calculated the Jacobian and null space of the Jacobian but with each row having a norm of 1, which we refer to as normalized Jacobian and null space of the Jacobian matrices, $\hat{J}\left(\bar{\boldsymbol{\theta}}\right)$ and $\widehat{J_{\text{N}}}\left(\bar{\boldsymbol{\theta}}\right)$ respectively. Then, at every training epoch, we obtained two 2D random variables from bivariate normal distributions, one for the task-potent noise and the other for the task-null noise, $\boldsymbol{n}_{\text{TP}}$ and $\boldsymbol{n}_{\text{TN}}$ respectively. Both random variables had a mean vector of 0, and their covariance matrices were the identity matrix times the variance in the task-potent space for $\boldsymbol{n}_{\text{TP}}$ and the variance in the task-null space for $\boldsymbol{n}_{\text{TN}}$, according to the combination of noise variance of the simulation. The motor command used in the training of the forward model is then defined by:

$$\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} + \boldsymbol{\theta}_{\text{n}} = \bar{\boldsymbol{\theta}} + \hat{J}\left(\bar{\boldsymbol{\theta}}\right)^T \boldsymbol{n}_{\text{TP}} + \widehat{J_{\text{N}}}\left(\bar{\boldsymbol{\theta}}\right)^T \boldsymbol{n}_{\text{TN}}. \tag{31}$$

The forward models were then trained for a total of 50 epochs, with a learning rate of 0.05, and at every epoch we evaluated the training using a batch of 50 motor commands obtained by adding the motor command $\bar{\boldsymbol{\theta}}$ with random samples from a multivariate normal distribution with a mean vector of 0 and standard deviation of 0.1 on all four axes, from which we calculate a test cost (Fig. 7A). The number of epochs and the learning rate used in the training of the forward models is insufficient to fully train them (insufficient to fully reduce their task-null components). This is intentional, however, because it creates a relationship between the accuracy of the forward model (how small their task-null component is) and the amount of variability, specifically task-null variability. We believe that this relates directly to one of the main findings of Singh et al. (2016), who reported that the nondominant hand exhibited smaller task-null variability and a lower estimated learning rate in the task, compared with the dominant hand, and the relationship between task-null variability and estimated learning rate was significant across participants using their dominant hands.

After the training of the forward models, we used them to train the inverse model (previously trained for 1000 epochs) under a visuomotor rotation of 45° for 500 epochs, with a learning rate of 0.1, again using the target $\boldsymbol{y}_{\text{tgt}} = \begin{bmatrix} 0 & 0.3 \end{bmatrix}^T$ as a desired outcome during training, and measuring the perpendicular distance between the inverse model outcome and the straight line connecting the start position $\boldsymbol{y}_0$ to the target $\boldsymbol{y}_{\text{tgt}}$ (Fig. 6F). After the visuomotor adaptation training, we trained the inverse models one more time, this time in the original environment (with the visuomotor rotation removed), to observe the training and the error during the postadaptation phase (Fig. 6F). Both trainings occurred 20 times for each combination of noise variance used to train the forward models.

After training of the inverse model, we analyzed the estimated learning rate during the visuomotor adaptation phase in the same way, using the same method as that in the experiment, by fitting the exponential function of Eq. (29) to the perpendicular error measured during the training. The average estimated learning rates across the 20 repetitions for each combination of noise variance were then compared with the amount of variance used in both task-null (Fig. 6G) and task-potent spaces (Fig. 6H) used in the training of the forward models.

Finally, we analyzed the training of the forward models by calculating their projections on both task-potent and task-null spaces after training (Fig. 7B). We calculated these projections using the Jacobian $J\left(\bar{\boldsymbol{\theta}}\right)$ and the null space of the Jacobian $J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)$ matrices using the equations:

$$\text{proj}_{\text{TP}} = \left\| \frac{\text{diag}\left(J\left(\bar{\boldsymbol{\theta}}\right)\widehat{\boldsymbol{M}}^T\right)}{\text{diag}\left(J\left(\bar{\boldsymbol{\theta}}\right)J\left(\bar{\boldsymbol{\theta}}\right)^T\right)} \right\|, \tag{32}$$

$$\text{proj}_{\text{TN}} = \left\| \frac{\text{diag}\left(J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)\widehat{\boldsymbol{M}}^T\right)}{\text{diag}\left(J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)J_{\text{N}}\left(\bar{\boldsymbol{\theta}}\right)^T\right)} \right\|, \tag{33}$$

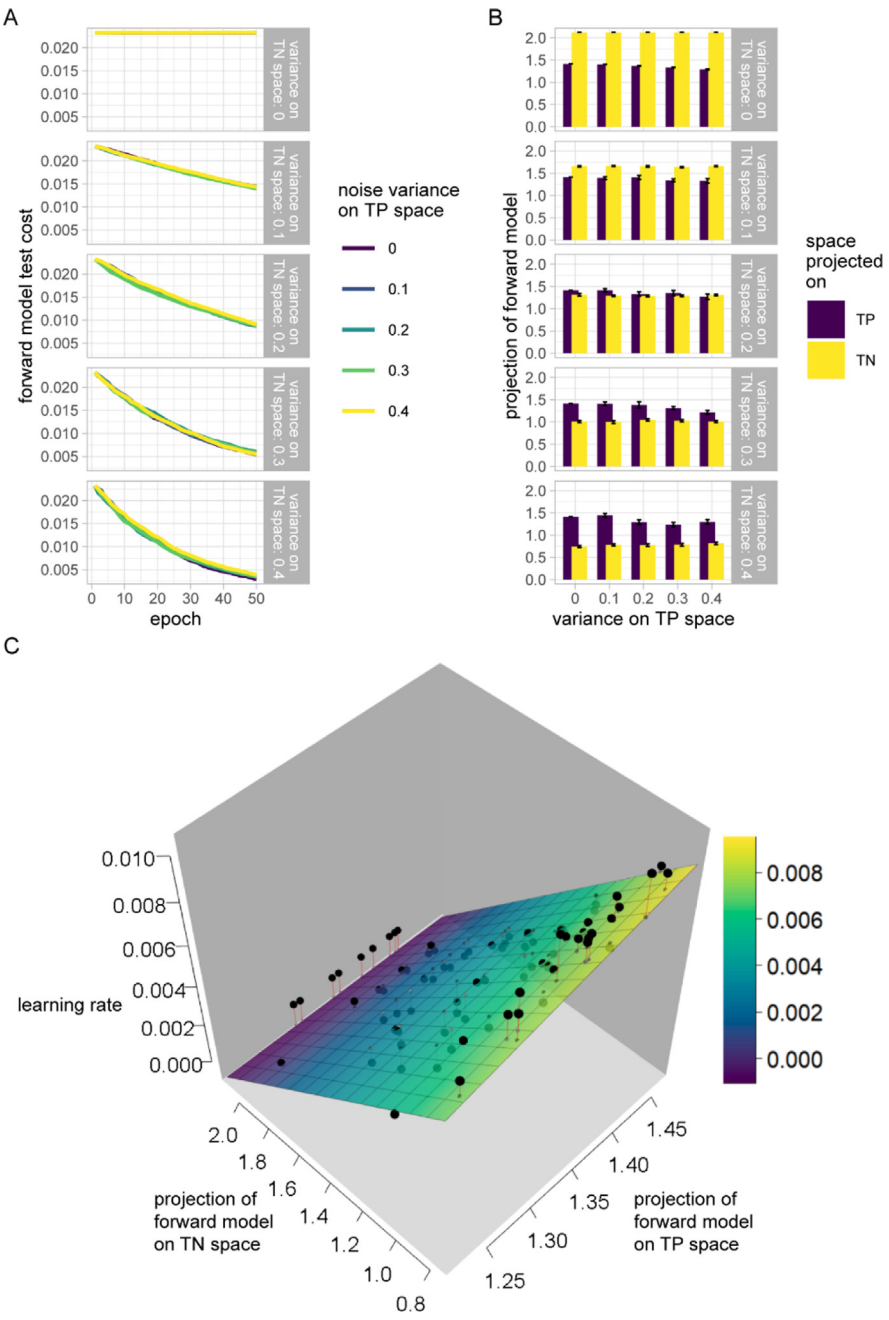

Fig. 7. Results from the simulations of the task from Singh et al. (2016). (A) Test cost of the forward models across the training epochs (training of the forward model occurred before the training of the inverse model throughout the experiment) for various combinations of motor noise in the task-potent (colors) and task-null (rows) spaces of motor commands. (B) Projection of the forward models (at the end of their training) on both task-potent and task-null spaces, for various combinations of motor noise in the task-potent (x axis) and task-null (rows) spaces of motor commands. Error bars represent one standard error. (C) Estimated learning rates (Z axis) of the inverse models (calculated in the same way as in the original experiment) trained by forward models with different amounts of projection in the task-potent and task-null spaces (X and Y axes). The forward models depicted here are the same as those in (A) and (B). Colored surface calculated with multiple linear regression with an interaction term. Data from the simulations (larger black spheres) are connected to the data predicted by the regression (smaller gray spheres) with thin lines, for better visualization. Lines, columns, and points in panels (A), (B), and (C) respectively are averages across 20 simulations. TP: task-potent. TN: task-null. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where proj_{TP} and proj_{TN} are the projections of the forward model on the task-potent and task-null spaces, respectively, $\widehat{\boldsymbol{M}}$ is the forward model at the end of training, and diag returns a vector with the elements in the diagonal of its matrix argument. These projections were then averaged across the 20 repetitions for each combination of noise variance, and compared across these variances (Fig. 7B). We also analyzed the estimated learning rate of the inverse models in terms of the projections of the forward models used to train them (Fig. 7C).

2.4. Hand gesture task

2.4.1. Task summary

The next task we simulated was reported in an experiment by Dal'Bello and Izawa (2021) (Fig. 8). In this experiment, an arbitrary mapping between hand gestures (measured with a data glove device detecting the joint angles of 10 joints of the hand) and cursor directions on a screen was defined by applying principal component analysis to a calibration dataset (Fig. 8A). The

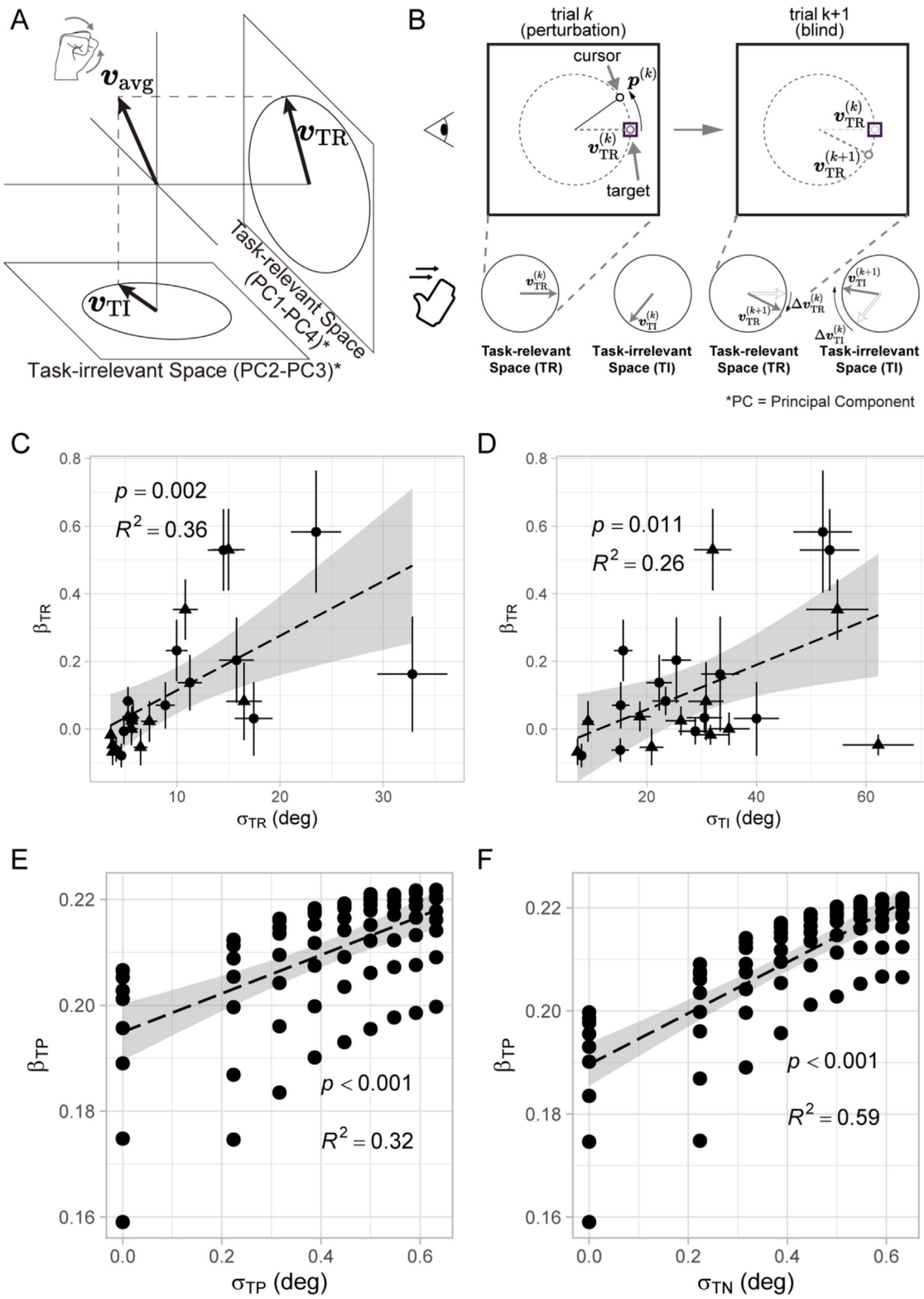

Fig. 8. Diagrams and results of the task from Dal'Bello and Izawa (2021). (A) In the task, participants made fast movements with their hands, and the hand average velocity vector in a trial, $\boldsymbol{v}_{avg}$ (composed of the angular velocity of 10 hand joints), was projected on two perpendicular 2D spaces, with the projection in the task-relevant space, $\boldsymbol{v}_{TR}$, being shown on the screen and therefore reflecting task error and reward, and the projection in the task-irrelevant space, $\boldsymbol{v}_{TI}$, not shown to participants. (B) Learning was analyzed in sequences of probe trials to a single target, where a visuomotor rotation $\boldsymbol{p}^{(k)}$ of either $+30°$, $-30°$, or $0°$ was applied to the cursor direction at trial k, and the change in motor commands between trials k and $k+1$ ($\Delta \boldsymbol{v}_{TR}^{(k)}$ and $\Delta \boldsymbol{v}_{TI}^{(k)}$ for the change in the task-relevant and task-irrelevant spaces, respectively) was measured. (C–D) Correlation between the estimated error-based learning rate β_{TR} and the variabilities in the task-relevant (C) and task-irrelevant (D) spaces, across participants in the experiment. σ_{TR}: standard deviation of noise in the task-relevant space. σ_{TI}: standard deviation of noise in the task-irrelevant space. (E–F) Correlation between estimated error-based learning rate β_{TP} and the variabilities in the task-potent (E) and task-null (F) spaces in our simulations of the motor task. Each point is the average of 20 simulations. σ_{TP}: standard deviation of noise in the task-potent space (equivalent to the experiment's task-relevant space). σ_{TN}: standard deviation of noise in the task-null space (equivalent to the experiment's task-irrelevant space). Panels A, B, C, D are reproduced from Dal'Bello and Izawa (2021).

orthogonal principal components were then used to define a pair of orthogonal, 2D control spaces: a task-relevant space, whose cursor direction was shown to participants on the screen and was therefore related to task error and reward, and the task-irrelevant space, whose cursor direction was not shown to participants and was not related to task error or reward.

In their task, learning was analyzed in sequences of probe trials (Fig. 8B), implemented after participants became skilled in the task. In the first trial of the sequence, a random perturbation between $+30°$, $-30°$ or $0°$ was added to the displayed cursor direction, and in the second trial of the sequence, the same perturbation was added but with an invisible cursor, so that reward would be higher if the participant changed their gesture to correct the error observed in the previous trial. The data from these probe sequences were then analyzed with a model of update of a motor memory, where the contribution of error-based learning in their model was defined as a change to the motor memory $\boldsymbol{x}$ (with components related to the reaching direction in both task-relevant and task-irrelevant spaces) between the first trial k of the probe sequence and the second trial $k + 1$ of the sequence proportional to the scalar angle error $e^{(k)}$ observed at trial k, such that:

$$\Delta \boldsymbol{x}_{\mathrm{EL}}^{(k)} = \begin{bmatrix} \beta_{\mathrm{TR}} \\ \beta_{\mathrm{TI}} \end{bmatrix} e^{(k)} \tag{34}$$

where β_{TR} and β_{TI} are the estimated learning rates of the error-based learning in the task-relevant and task-irrelevant spaces, respectively.

In their experiment, a significant, positive correlation was found between the estimated task-relevant, error-based learning rate β_{TR} and the estimated motor variabilities in both task-relevant (Fig. 8C) and task-irrelevant (Fig. 8D) spaces. Further modeling results showed that these effects were simultaneous, and not mediated by one another, suggesting a role of the exploration of both task-relevant and task-irrelevant motor commands on error-based motor learning. On the basis of our Predictions 2, 3 and 4, we interpret these results as reflecting the exploration of the entire motor command space leading to the learning of an efficient sensitivity derivative, which results in greater error-based learning during the sequences of probe trials, with a larger amount of exploration reflecting a faster convergence of sensitivity derivatives to the ideal value.

2.4.2. Simulations

To simulate this task using our framework, we first defined an arbitrary linear map between 10D motor commands and 2D task outcomes by initializing a random 10D base vector (with random variables from a normal distribution with a mean of 0 and a standard deviation of 1), then obtaining a family of orthogonal vectors covering the entire 10D motor command space. From this family of vectors, we arbitrarily selected one pair of vectors for our task-potent (referred to as task-relevant in the previous study) space, and a separate pair of vectors for our task-null (referred to as task-irrelevant in the previous study) space of motor commands, assembling 2×10 environment matrices $\boldsymbol{M}_{\mathrm{TP}}$ and $\boldsymbol{M}_{\mathrm{TN}}$, respectively. In all simulations, the linear forward models were initialized as $\widehat{\boldsymbol{M}} = 0.5\boldsymbol{M}_{\mathrm{TP}} + 0.5\boldsymbol{M}_{\mathrm{TN}}$ and the inverse models were initialized as $\boldsymbol{W} = \boldsymbol{M}_{\mathrm{TP}}^{T} + \boldsymbol{M}_{\mathrm{TN}}^{T}$. This procedure was conducted so that the forward models started with a component in both the task-potent and task-null spaces, and so that the inverse models started with good task performance, being able to generate appropriate motor commands for any targets in the 2D task space.

Each simulation started with the training of the forward model for 100 epochs. At each training epoch, a random exploration noise signal $\boldsymbol{u}_{\mathrm{n}}$ was added to a default motor command $\boldsymbol{u}_{\mathrm{DEF}}$, initialized by presenting the desired outcome $\boldsymbol{y}^{*} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{T}$ to the inverse model and obtaining its output ($\boldsymbol{u}_{\mathrm{DEF}} = \boldsymbol{W}\boldsymbol{y}^{*}$). We first obtained two 2D random variables from bivariate normal distributions, one for the task-potent noise and the other for the task-null noise, n_{TP} and n_{TN} respectively, both with a mean vector of 0, and their covariance matrices were the identity matrix times the variance in the task-potent space, σ_{TP}^{2}, for n_{TP} and the variance in the task-null space, σ_{TN}^{2}, for n_{TN}, according to the combination of the noise variance of the simulation. The motor command used in the training of the forward model at that epoch was then defined by:

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{DEF}} + \boldsymbol{u}_{\mathrm{n}} = \boldsymbol{u}_{\mathrm{DEF}} + \boldsymbol{M}_{\mathrm{TP}}^{T} n_{\mathrm{TP}} + \boldsymbol{M}_{\mathrm{TN}}^{T} n_{\mathrm{TN}}, \tag{35}$$

its outcome $\boldsymbol{y}$ in the task space was measured, and the pair $(\boldsymbol{u}, \boldsymbol{y})$ was used as a training sample for the forward model, with a learning rate of 0.05.

After the training of the forward model, we implemented single-epoch trainings of the inverse model under perturbations. For each of the three perturbations of $+30°$, $-30°$, and $0°$, the perturbation was added to the task outcome $\boldsymbol{y}(k)$ obtained from the motor command generated by the inverse model when presented with the desired outcome $\boldsymbol{y}^{*} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{T}$, resulting in the perturbed task outcome $\boldsymbol{y}(k)_{\mathrm{perturb}}$. Learning of the inverse model then proceeded for one epoch by backpropagating the error $\boldsymbol{y}(k)_{\mathrm{perturb}} - \boldsymbol{y}^{*}$ through the forward model (acting as a sensitivity derivative). After the single epoch of update, we measured the task outcome $\boldsymbol{y}(k + 1)$ of the motor command generated by the inverse model with the presentation of the same desired outcome as before. Because the neural network is deterministic, we only needed a single data point from each amount of perturbation. Having obtained the change in angle between the task outcomes at trials $k + 1$ and k, $\Delta\theta\,(k)$, **w**ith the function $\Delta\theta\,(k) = \mathrm{atan2}\,(\boldsymbol{y}\,(k)\,, \boldsymbol{y}\,(k+1))$, and having the angle of each perturbation $\theta\,(k)_{\mathrm{perturb}}$, we then fit the following linear function to the three data points from the three perturbations:

$$\Delta\theta\,(k) = \beta_{\mathrm{TP}}\theta\,(k)_{\mathrm{perturb}}\,, \tag{36}$$

where β_{TP} is the estimated learning rate obtained as a consequence of the training of the forward model.

Our simulations varied the amount of variance in the task-potent and task-null spaces of motor commands, σ_{TP}^{2} and σ_{TN}^{2} respectively, among the nine possible values ranging from 0.0 to 0.4, at every 0.05, totaling 81 possible combinations of noise variances in both spaces. 20 simulations were conducted for each combination of variances. We then averaged the estimated learning rates β_{TP} across all simulations for each combination of variances and compared them with the standard deviation of the exploration noise used in the simulation in both task-potent and task-null spaces (σ_{TP} and σ_{TI}, respectively) (Fig. 8E–F). We also analyzed the test cost of the forward models throughout training (Fig. 9A) using a batch of 50 motor commands generated from a multivariate normal distribution with a mean vector of 0 and a standard deviation of 0.5 in each of the 10 axes. We also measured the projection of the forward models on both task-potent and task-null spaces (Fig. 9B) for each combination of variances, using a similar procedure to that performed in the arm reaching task simulations, but using the environment matrices $\boldsymbol{M}_{\mathrm{TP}}$ and $\boldsymbol{M}_{\mathrm{TN}}$ in place of matrices $J\left(\overline{\theta}\right)$ and $J_{\mathrm{N}}\left(\overline{\theta}\right)$ in Eq. (32) and Eq. (33), respectively. In addition, we further compared the estimated learning rates β_{TP} of the inverse models with the projection of the forward models on the task-potent and task-null spaces (Fig. 9C).

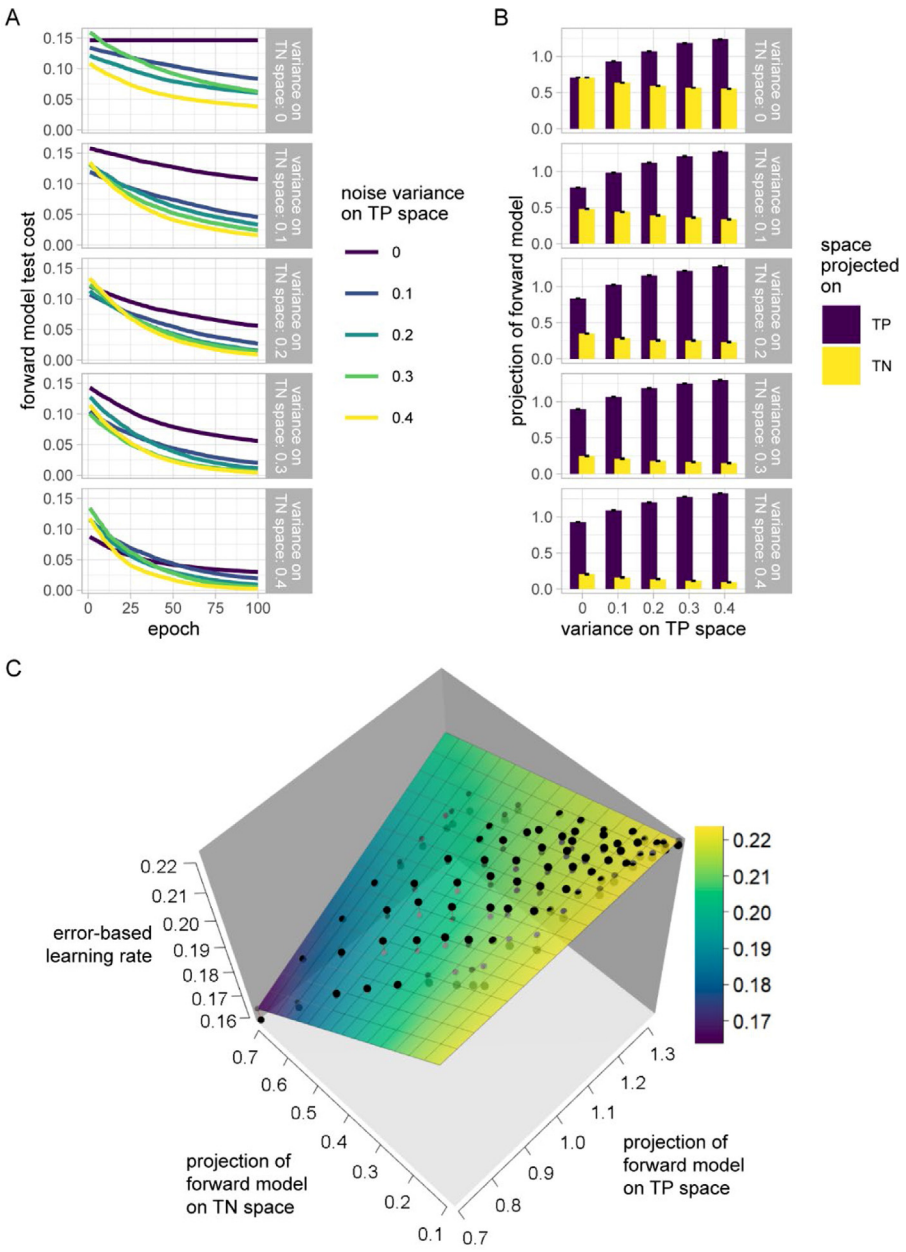

Fig. 9. Results from the simulations of the task from Dal'Bello and Izawa (2021). (A) Test cost of the forward models across the training epochs for various combinations of motor noise in the task-potent (colors) and task-null (rows) spaces of motor commands. (B) Projection of the forward models (at the end of training) on both task-potent and task-null spaces, for various combinations of motor noise in the task-potent (x axis) and task-null (rows) spaces of motor commands. Error bars represent one standard error. (C) Estimated error-based learning rates (z axis) of the inverse models trained by forward models with different amounts of projection in the task-potent and task-null spaces (*x* and *y* axes), obtained by applying instantaneous perturbations in the task, similar to the method performed in the experiment. The forward models depicted here are the same as those in (A) and (B). Colored surface calculated with multiple linear regression with an interaction term. Data from the simulations (larger black spheres) are connected to the data predicted by the regression (smaller gray spheres) with thin lines, for better visualization. Lines, columns, and points in panels (A), (B), and (C), respectively, are averages across 20 simulations. TP: task-potent. TN: task-null. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.5. EMG task

2.5.1. Task summary

The final task we simulated was an experiment reported by Berger et al. (2013). In this experiment, participants used the activity of 13 muscles in the arm and torso to control the direction of a cursor on a screen. The mapping between muscle activity and cursor direction was initially calculated by applying multiple linear regressions to the EMG signals recorded from the 13 muscles while participants generated forces in eight different directions, obtaining the 2×13 linear mapping matrix $\boldsymbol{H}$, with each column representing the estimated force direction of each muscle (Fig. 10A). The virtual force used to control the cursor was then calculated as:

$$\boldsymbol{f} = \boldsymbol{H}\boldsymbol{m}, \tag{37}$$

where $\boldsymbol{f}$ is the 2D virtual force, and $\boldsymbol{m}$ is the 13D muscle activation signal extracted from the EMG of the 13 muscles used in the experiment.

In addition to calculating the muscle activity to force mapping, muscle synergies were obtained by applying nonnegative matrix factorization to the initial dataset of muscle activity, resulting

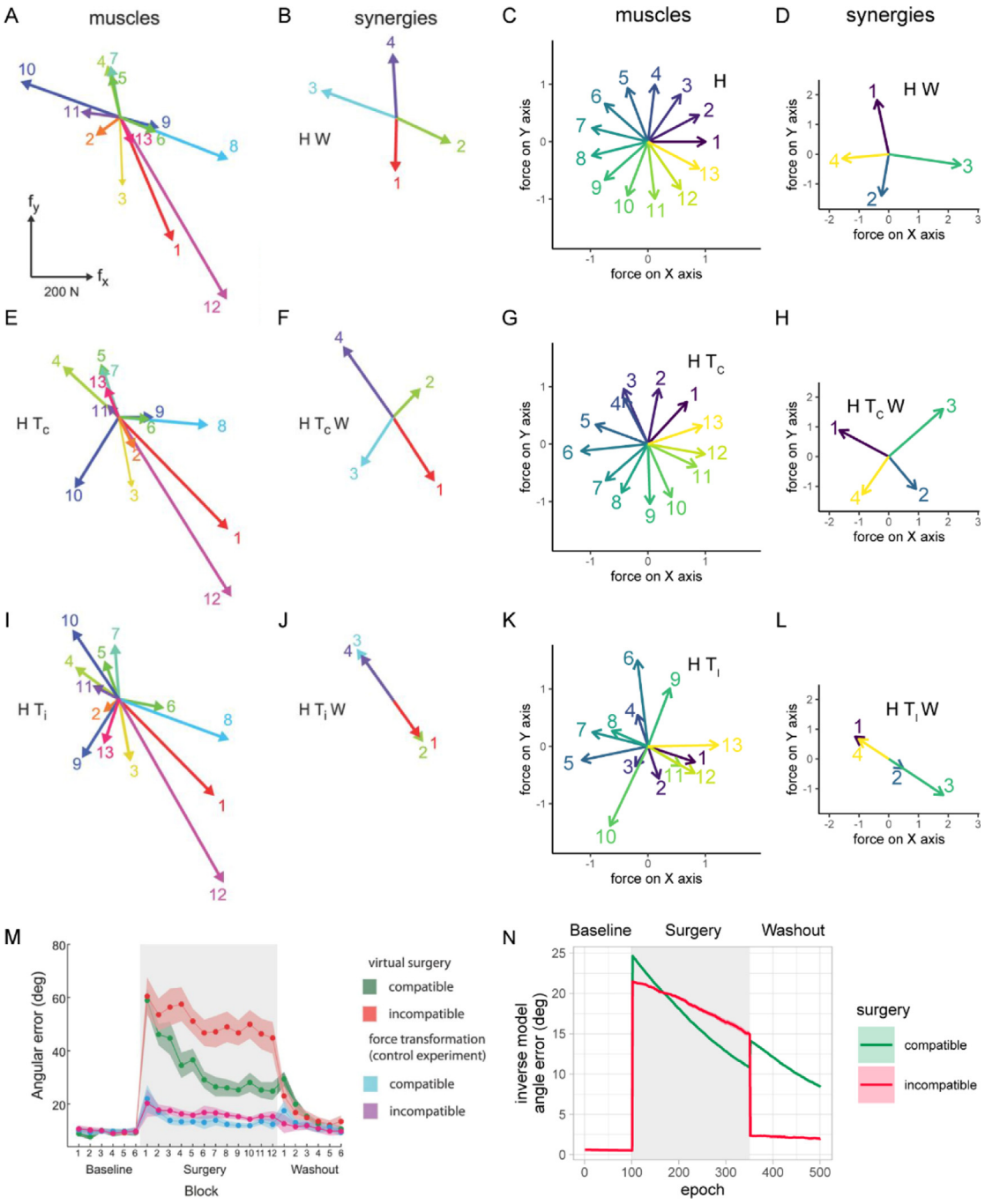

Fig. 10. Diagrams and results of the task from Berger et al. (2013). In their task, the surface EMG from 13 muscles was measured and a mapping between muscle activation and hand force was calculated (matrix H), so that participants could control the direction of a cursor using this estimated force instead of the actual force. During an initial force control task, non-negative matrix factorization was used on the EMG data to identify synergies for each participant (matrix W). Panels to the left (A, B, E, F, I, J, M) are from the original experiment and were reproduced from Berger et al. (2013). Panels to the right (C, D, G, H, K, L, N) are from our simulations. (A) EMG-to-force mapping matrix H from a representative participant in the experiment. Each arrow represents the estimated 2D force generated by each muscle during its activation. (B) Muscle synergies matrix W from the participant shown in (A) mapped to 2D force vectors. (C) EMG-to-force mapping matrix H used in all our simulations. (D) Muscle synergies matrix W for the EMG data from an inverse model trained to reach for targets using the muscles with the EMG-to-force mapping H of (C). (E) Forces generated by each muscle after the application of a compatible surgery represented by the surgery matrix T_C, for the same participant shown in (A). (F) Muscle synergies mapped to force after the application of the compatible surgery shown in (E). (G) Forces generated by each muscle in our simulations after the application of a compatible surgery. (H) Muscle synergies mapped to force in our simulations after the application of the compatible surgery shown in (G). (I) Forces generated by each muscle after the application of an incompatible surgery represented by the surgery matrix T_I, for the same participant shown in (A). (J) Muscle synergies mapped to force after the application of the incompatible surgery shown in (I). (K) Forces generated by each muscle in our simulations after the application of an incompatible surgery. (L) Muscle synergies mapped to force in our simulations after the application of the incompatible surgery shown in (K). (M) Angular error between initial movement direction and target direction across participants in each block and for each surgery, including force transformations (not analyzed in our simulations). Solid lines and shaded regions represent means and standard errors, respectively. (N) Angular error between task outcome of the motor commands generated by the inverse models and targets throughout training, across both surgeries. Solid lines and shaded regions represent means and standard errors across 20 simulations, where motor noise with a variance of 0.1 in the task potent space and of 0.05 in the task-null space (the original task-potent and task-null spaces, during the baseline) was added to the motor commands used in the update of both forward and inverse models simultaneously.

in the 13 × s nonnegative matrix $\boldsymbol{W}$, where s is the number of synergies calculated for that particular participant. Each column in the $\boldsymbol{W}$ matrix represents the relative recruitment of each of the 13 muscles by that specific synergy. A muscle activity vector $\boldsymbol{m}$ could then be partially reconstructed by the synergies as:

$$\boldsymbol{m} = \boldsymbol{W}\boldsymbol{c}, \tag{38}$$

where $\boldsymbol{c}$ is an s-D vector of synergy recruitment coefficients. In conjunction with the muscle activity to force matrix $\boldsymbol{H}$, each synergy also has a force in the 2D task space (Fig. 10B).

The authors then investigated participants' learning during the application of a "virtual surgery", which changed the direction of the forces generated by each muscle in the virtual environment. Two types of surgery were defined based on the capacity of the previously calculated synergies to span the entire task space: in compatible surgeries, the muscle synergies still spanned the entire task space (Fig. 10E, F); in incompatible surgeries, the synergies did not span the entire task space (Fig. 10I, J). The surgeries were defined in terms of transformation matrices ($\boldsymbol{T}_c$ for the compatible surgery and $\boldsymbol{T}_i$ for the incompatible surgery), which multiplied $\boldsymbol{H}$ in Eq. (37), causing a rotation in the virtual force generated by each muscle. The authors' main finding was that the learning (analyzed with, among other metrics, the angular error between the initial movement direction and the target direction) during compatible surgeries was faster than that during incompatible surgeries (Fig. 10M). In addition, synergies calculated from participants at the end of the incompatible surgery phase were different from those extracted before the surgeries, suggesting that during the incompatible surgery the participants started to form new synergies. The authors concluded that muscle synergies were likely to act as control modules during the generation of force, and that virtual surgeries which required a change in the relationship between synergies/control modules and force direction were learned faster than surgeries which required the learning of new synergies/control modules, which was argued to be a slower process.

Our interpretation of Berger et al.'s (2013) findings focused on which motor command spaces are explored during the learning of a new sensitivity derivative for each virtual surgery. From our Predictions 1 and 3, we believe that the reason participants adapted to the compatible virtual surgery faster was that their motor exploration occurred mostly in the dimensions explained by the synergies, which still span the entire task-potent space in the compatible surgery, whereas in the incompatible surgery, in which non-synergy motor commands are now part of the task-potent space, exploration of this new task-potent space is smaller, resulting in slower learning.

2.5.2. Simulations

For the simulations of this experiment, instead of training the forward model and then analyzing the training of the inverse model, as in the simulations of the two previous studies, we decided to train both forward and inverse models simultaneously, similarly to the procedure used in simulation 2 of the simple linear motor task, which we believe is more similar to the way in which motor commands would be generated in a biological system. We also sought to confirm whether a neural network architecture that better replicates the property of directional tuning observed in neurons of the motor system (Georgopoulos et al., 1986) was able to replicate the results of the experiment. To that end, we used neural networks with radial basis functions (Thoroughman & Shadmehr, 2000) in both our forward and inverse models. In the case of the inverse models, whose inputs are 2D desired outcomes, we defined 17 basis functions for each of the two axes of the inputs, so that for a desired outcome $\boldsymbol{y}^* = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$, the output of the kth basis function on the first axis of the desired outcome is given by:

$$g_k(y_1) = \exp\left(-\frac{(y_1 - c_k)^2}{2\sigma^2}\right), \tag{39}$$

where c_k is the center of the radial basis function in that axis and σ is the standard deviation of the basis function, and equivalently for the second axis of the desired outcome, totaling a 34D output of this basis function layer. This layer is then followed by a sigmoidal layer, whose weights are then modified with training. The sigmoidal layer limits the output of the network to 13D (the number of muscles used in the task), and to be between 0 and 1, as was the muscle activity recorded from the EMG during the experiment. The basis function centers were uniformly spread in a region between −2 and +2, with $\sigma = 0.175$. Similarly, in the forward models, we defined 16 basis functions for each axis of their input, in this case, a 13D motor command, resulting in a 13 × 16 radial basis function layer output, which is then followed by a linear layer which reduces the dimension of the output to 2D, the same as the task space. In the forward models, the centers of the basis functions were uniformly spread in a region between −0.5 and +1.5 (covering the region from 0 to 1 of the motor commands), with $\sigma = 0.1$.

We defined an initial mapping between muscle activation and virtual force $\boldsymbol{H}$ with the force generated by the 13 muscles being evenly spread around the task space (Fig. 10C). Then, we initially trained the inverse model for 200 epochs, with a learning rate of 0.01, and using the true sensitivity derivative during the update, so that the inverse models at the beginning of the virtual surgeries had a low error. For this phase of training, we used a batch of 20 desired outcomes, uniformly distributed along a circle with radius 1.5 around the origin of the task space. After this training, we then trained the forward model for 500 epochs with a learning rate of 0.1, using as targets 500 motor commands obtained from a multivariate uniform distribution with limits between 0 and 1.

After this initial training, we calculated synergy coefficients of the motor commands generated by the inverse model, using a grid of 121 points as desired outcomes, uniformly spanning the region between −1.5 and +1.5 in both axes of the task space. We then used these synergy coefficients to calculate the transformation matrices $\boldsymbol{T}_c$ and $\boldsymbol{T}_i$ for the compatible (Fig. 10G, H) and incompatible (Fig. 10K, L) virtual surgeries, respectively, using the same methods as were used in the original experiment. Our criterion to define if a pair of surgeries were valid or not was based on whether both surgeries generated similar test costs for both inverse and forward models trained earlier. In addition, a minimum average absolute error of 20° in the motor commands generated by the inverse models during both surgeries, using eight targets uniformly distributed along a circle with a radius of 1.5 as desired outcomes, was a necessary condition for the validity of the surgeries. For the test cost of the inverse model, we used a batch of 20 desired outcomes, uniformly distributed along a circle with a radius of 1.5 around the origin of the task space, as was used during the initial training. For the test cost of the inverse model, we used the same test batch of motor commands used as a test in the initial training.

With the surgeries defined, both forward and inverse models were trained for 250 epochs simultaneously under the two virtual surgeries, and with different combinations of exploration noise variance in task-potent and task-null spaces, with task-potent and task-null referring to the original, unperturbed environment. The addition of exploration noise was performed in a similar way as in simulation 2 of the simple linear motor task. At each training epoch, one target out of eight possible targets (uniformly distributed along a circle with a radius of 1.5) was selected, and a motor command $\tilde{\boldsymbol{m}}$ was obtained using the target as the input of the inverse model. Having calculated orthonormal basis

$\boldsymbol{M}_{\mathrm{TP}}$ and $\boldsymbol{M}_{\mathrm{TN}}$ for the original task-potent and task-null spaces, respectively, with $\boldsymbol{M}_{\mathrm{TP}}$ being 2×13 and $\boldsymbol{M}_{\mathrm{TN}}$ being 11×13, we obtained two random variables from multivariate normal distributions, one for the task-potent noise (2D, the dimension of the task space) and the other for the task-null noise (11D, the dimension of the null space), $\boldsymbol{n}_{\mathrm{TP}}$ and $\boldsymbol{n}_{\mathrm{TN}}$ respectively, both with a mean vector of 0. Their covariance matrices were the identity matrix (2×2 for the task-potent noise and 11×11 for the task-null noise) times the variance in the task-potent space, σ^2_{TP}, for $\boldsymbol{n}_{\mathrm{TP}}$ and the variance in the task-null space, σ^2_{TN}, for $\boldsymbol{n}_{\mathrm{TN}}$, according to the combination of noise variance of the simulation. The noisy motor command $\boldsymbol{m}$ used in the training of both forward and inverse models was then defined as:

$$\boldsymbol{m} = \bar{\boldsymbol{m}} + \boldsymbol{m}_{\mathrm{n}} = \bar{\boldsymbol{m}} + \boldsymbol{M}^T_{\mathrm{TP}} n_{\mathrm{TP}} + \boldsymbol{M}^T_{\mathrm{TN}} n_{\mathrm{TN}}. \tag{40}$$

Then, as was performed in simulation 2 of the simple linear motor task, $\boldsymbol{m}$ was used as a training motor command for the forward model, and at the same epoch, the inverse model was also updated, with the error used in the update calculated with the selected target and the outcome of the motor command $\boldsymbol{m}$, under the virtual surgery being simulated. The learning rates of this training under the virtual surgeries were 0.02 for both inverse and forward models, and there were 20 repetitions of each combination of noise variance in the task-potent and task-null spaces, with nine combinations in total, with the noise variance in one space varying between 0, 0.05, and 0.1. After training under the virtual surgeries, we trained both the forward and inverse models again with the virtual surgeries removed, with the same parameters as before, for 200 epochs.

To analyze the training of both forward and inverse models under the different combinations of exploration noise in the original task-potent and task-null spaces, under both compatible and incompatible virtual surgeries, we calculated the average absolute angle error of the motor commands generated by the inverse model to eight targets uniformly distributed in a circle with a radius of 1.5 (Fig. 10N). We also compared the test cost of both the forward and inverse models calculated during the training under the virtual surgeries across all combinations of noise variance in the original task-potent and task-null spaces (Fig. 11).

3. Results

3.1. Simple linear motor task

In simulation 1 of the simple linear motor task, where the training of both forward and inverse models happened simultaneously, the results revealed that a non-zero exploration noise in the entire motor command space enabled the forward model to converge to the correct solution (Fig. 4A) (our Prediction 4), with a larger amount of noise leading to a faster convergence (our Prediction 3). In addition, we found that exploration noise only in the original task-potent space, which became the task-null space, only contributed to reducing the task-null component of the forward model, and exploration noise only in the original task-null space, which became the task-potent space, contributed to bringing the task-potent component of the forward model to its correct value of 1, as was predicted in our earlier analyses of the update equations. The results revealed that whenever there was no exploration in one of the motor command spaces, the forward model's test cost did not fully decrease during training (Fig. 4B). In terms of its function as a sensitivity derivative, when the forward model maintained a significant task-null component (while its task-potent component converges to the correct value), such as when there was no task-null exploration, we can see that, while the inverse model was updated towards a correct solution (Fig. 4C) (our Prediction 1), this update was slower than when the forward model's task-null component was smaller (Fig. 4D) (our Prediction 2). This occurred because of the limitation in the amplitude of the motor correction signal used in the update of the inverse model. As described above, with such limitations it is desirable to reduce the task-null component of the motor correction to maximize the task-potent component, enabling a faster update of the inverse model. The update of the inverse model also fails when there is no exploration in the task-potent (original task-null) space, because the forward model has a zero task-potent component.

In simulation 2, both forward and inverse models were updated simultaneously, but this time the motor command used to train the forward model was generated by the inverse model, and exploration noise was added to this motor command at every epoch. The results revealed that, even with no task-null space (original task-potent space) exploration, the forward model still reduced its task-null component (Fig. 5A), although greater exploration noise led to a faster reduction of this component (Fig. 5B). This is because the motor command generated by the inverse model had a non-zero task-null (original task-potent) component, therefore enabling the update of the forward model in the task-null space. Its task-potent component, however, only updated with non-zero task-potent (originally task-null) exploration noise (Fig. 5A, Fig. 5B) with a larger amount of exploration leading to a faster update, also enabling the appropriate update of the inverse model (Fig. 5C, D).

In summary, the simulations of this simple linear motor task exhibited many of the features that were found previously by analyzing the inverse and forward models' update equations. Such features included: (1) the update of the forward model occurred in the direction of the motor command used to train it; (2) exploring the entire motor command space enabled the complete training of the forward model, with the reduction of its task-null component leading to a faster update of the inverse model; and (3) the amount of exploration noise influenced the speed of the training.

3.2. Arm reaching task

In our simulations of the arm reaching task reported by Singh et al. (2016), we observed a similar reduction of task error during the visuomotor adaptation phase (Fig. 6F) compared with the experimental results (Fig. 6C), as well as a similar aftereffect when the visuomotor perturbation was removed. We also replicated Singh et al.'s (2016) main finding of a positive correlation between task-null variability and estimated learning rate during the visuomotor adaptation phase (Fig. 6G) and a non-significant correlation between task-potent variability and learning rate (Fig. 6H).

By analyzing the forward model test cost during its training (Fig. 7A), before its usage as a sensitivity derivative for the inverse models during the visuomotor adaptation and postadaptation phases, we found an effect of the amount of task-null variability on the speed of the decrease of the test cost (Prediction 3), but no clear effect of the amount of task-potent variability. This finding is likely to have been caused by the initial conditions of the forward models, which were initialized with large task-potent and task-null components. Under these initial conditions, the task-potent component of the forward models was correct and did not need to be updated. The results revealed that, after the 50 epochs of training, the projection of the forward models on the task-potent space was largely the same across all combinations of noise variance (Fig. 7B). However, their projection on the task-null space became smaller as the task-null exploration increased, reflecting the effects of the amount of task-null noise on the speed of the update before the forward models were fully trained.

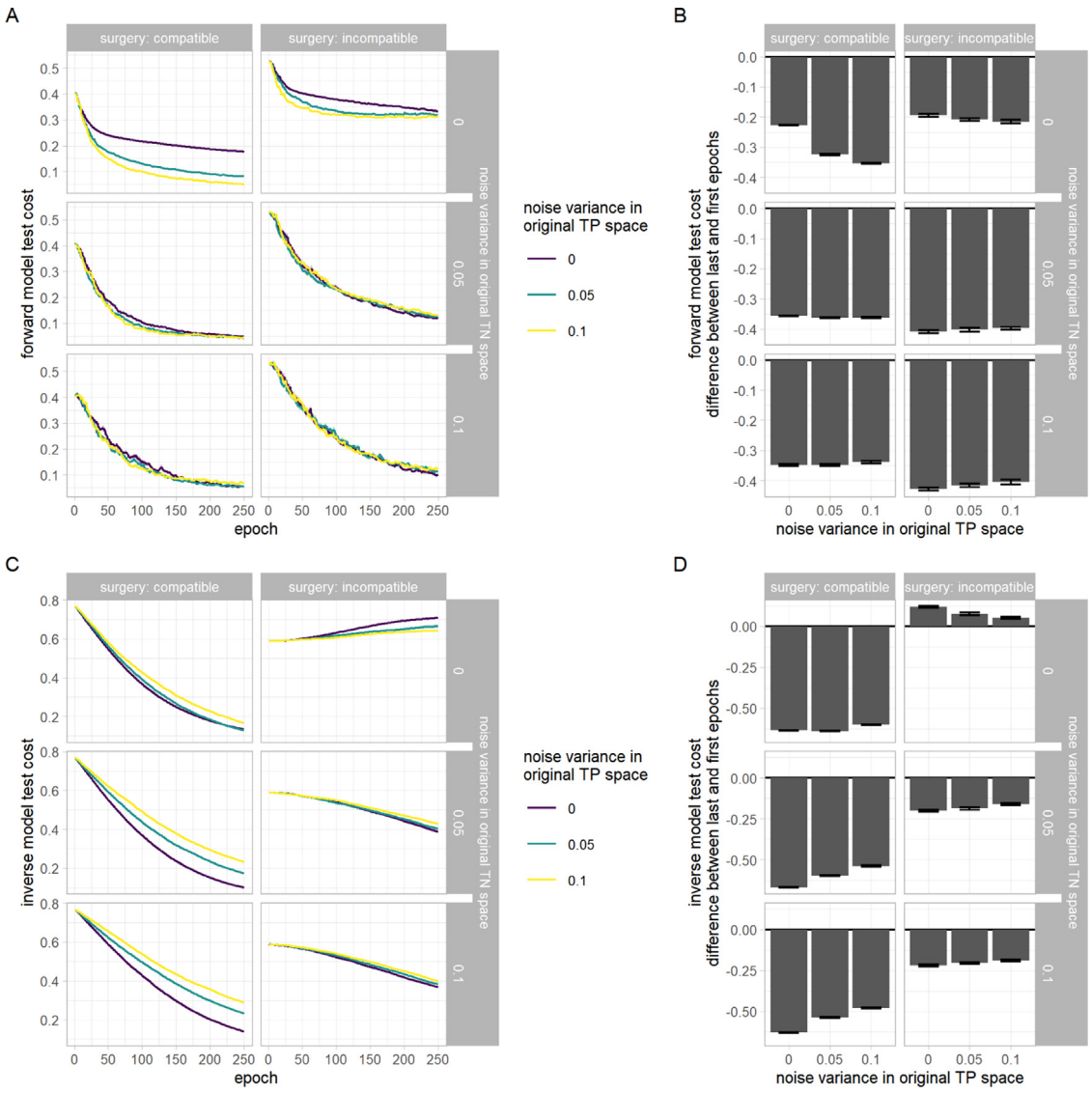

Fig. 11. Detailed results of the simulations of the task from Berger et al. (2013). (A) Test cost of the forward models across the training epochs for various combinations of motor noise in the original task-potent (colors) and task-null (rows) spaces of motor commands. (B) Difference of forward model test cost between the last and first epochs of training, for all combinations of motor noise. (C) Test cost of the inverse models across the training epochs (training occurred simultaneously with the forward model, and using the same motor command) for various combinations of motor noise in the original task-potent (colors) and task-null (rows) spaces of motor commands. (D) Difference of inverse model test cost between the last and first epochs of training, for all combinations of motor noise. Results for simulations using compatible and incompatible virtual surgeries are shown in the columns of each panel. Each line + shaded region in panels A and C and columns + error bars in panels B and D represent averages and standard errors across 20 simulations.

Ultimately, the projection of the forward models on the task-null space, which was shown to be related to task-null variability, has a large effect on the learning rate of the inverse models during the visuomotor adaptation phase (Fig. 7C), with less projection leading to a higher learning rate. This is because the limitation in the motor correction obtained from the sensitivity derivatives, with a smaller task-null component of the forward model leading to a larger task-potent component of the motor correction, resulted in an overall faster update of the inverse model (Prediction 2).

3.3. Hand gesture task

In our simulations of the hand gesture task reported by Dal'Bello and Izawa (2021), the results revealed that the amount of task-potent and task-null variability used to train the forward models had a significant effect on the estimated learning rate calculated from the inverse models during the single-trial perturbations (Fig. 8E, F). These results are similar to the findings reported in the original experiment (Fig. 8C, D). The decrease of the forward models' test cost during training was faster with a larger amount of exploration, in both task-potent and task-null spaces (Fig. 9A). This is likely to be because of the initial conditions of the forward models in the simulations: being a de novo learning task, we assumed that the learners had no initial knowledge of which motor commands are task-potent and task-null, and therefore we initialized the simulations' forward models with small task-potent and task-null components. The forward models then benefitted from both task-potent exploration, which increased their projection on the task-potent space, and from task-null exploration, which decreased their projection in the task-null space (Fig. 9B) (Prediction 4). Because of the limitation in the amplitude of the motor correction generated by the forward models (acting as sensitivity derivatives),

the estimated learning rate of the inverse models during the single-trial adaptation had a positive correlation with the forward models' projection in the task-potent space, and a negative correlation with the forward models' projection in the task-null space (Fig. 9C).

3.4. EMG task

In our simulations of the experiment reported by Berger et al. (2013), in which both forward and inverse models were trained simultaneously, the graph of the change in angular error over training in the two virtual surgeries in the original experiment was replicated well (Fig. 10M), with a combination of variances of 0.1 in the original task-potent space and of 0.05 in the original task-null space of motor commands (Fig. 10N). The angular error for the inverse models trained decreased faster in the compatible virtual surgery compared with that in the incompatible surgery. In addition, there was a larger after-effect during the first block after the compatible surgery, compared with that after the incompatible surgery.

When analyzing the test cost of the forward models during the training under the compatible surgery (Fig. 11A), we can see that, when no task-null exploration is added, more exploration in the original task-potent space leads to a faster decrease in the test cost, and to a larger decrease at the end of training (Fig. 11B). However, in the presence of exploration in the original task-null space, the amount of task-potent space exploration does not seem to have an effect on either the speed or the amount of decrease in the test cost. The results revealed that, in all cases, the inverse models' test cost decreased substantially (Fig. 11C). However, this decrease appears faster, and larger, for smaller amounts of task-potent exploration (Fig. 11D). This is more evident when there is a non-zero task-null exploration. We believe that this is caused by the addition of motor noise to the motor command generated by the inverse model during training. In the simulations with no task-null exploration, this negative effect of task-potent noise is alleviated by the faster training of the forward model (Fig. 11A, B), used in the inverse models' update as a sensitivity derivative.

Under the incompatible surgery, when no task-null exploration was added, the forward model was not trained well, in contrast with the simulations with non-zero task-null exploration (Fig. 11A). This result is in accord with the finding that the inverse model's test cost during training did not decrease under this condition (Fig. 11C), and actually increased over training (Fig. 11D). With the addition of task-null exploration noise, the forward models' test cost decreased substantially, enabling the subsequent decrease of the inverse models' test cost.

We believe that the absence of task-null exploration leading to the failure of the training of both forward and inverse models was caused by the incompatible virtual surgery requiring motor commands which were not explored in the absence of task-null exploration. The incompatible surgery was formulated as requiring the usage of motor commands that were not encoded by the synergies initially used in the task. In our simulations, the motor commands generated by the inverse model, with exploration noise added to update the forward model, could be considered to be mainly composed of the synergies calculated previously. Because the synergies exhibited a substantial intersection with the task-potent space, the new environment thus required the usage of motor commands that were in the original task-null space. Although the synergies exhibited an intersection with the original task-null space, this intersection was relatively small (because our synergies encompassed a 4D subspace of motor commands, and our task-potent space was 2D, the intersection between the synergy space and the task-null space was a 2D subspace) compared with the entire task-null space, and did not encompass the necessary motor commands during the incompatible surgery, by its definition. Therefore, these motor commands were not explored, and the learning of the forward model failed. This situation is similar to that in simulation 2 of the simple linear motor task (Fig. 5), in which, even though the motor command generated by the inverse model alone, without the addition of exploration noise, reduced the forward model's task-null component, its training ultimately failed because task-potent (original task-null) motor commands were not explored.

In summary, the current results revealed that, in the incompatible surgeries used in this experiment, exploration of the original task-null space was essential for the appropriate training of a forward model, which, when used as a sensitivity derivative, enabled the training of the inverse model and the subsequent reduction of task error.

4. Discussion

In the simulations presented in the current study, we reproduced many of the results reported in three previous experiments in a range of motor tasks. We initially mathematically derived some of the properties of our proposed learning mechanism, including the characteristic that, under the assumption that the neural representation of a motor correction signal has a limited amplitude, the exploration of the entire motor command space leads to the convergence of a sensitivity derivative (represented here by a forward model) to a solution that enables the efficient training of an inverse model to generate motor commands in a novel motor task. This prediction was confirmed with simulations of a simple linear motor task. Our proposed learning mechanism was shown to replicate well the relationship between task-potent and task-null variabilities with the estimated learning rate observed in a de novo hand gesture task (Dal'Bello & Izawa, 2021). In an arm reaching task (Singh et al., 2016), the results showed that when the sensitivity derivative's projection in the task-null space was inversely related to the amount of task-null variability, a significant correlation was observed between task-null variability and estimated learning rate under a visuomotor perturbation. In an EMG task (Berger et al., 2013), the simulation results revealed that, when motor exploration was constrained to the originally recruited synergies, the learning of a controller (and of a sensitivity derivative) in an incompatible virtual surgery, which required the usage of motor commands not encoded by previously recruited synergies, was severely limited. Thus, these results provide evidence that the proposed learning mechanism might be implemented in the human brain and used during motor learning.

4.1. Elucidating the role of motor exploration on error-based de novo learning

Our proposed learning mechanism bridges the gaps between motor exploration (Sternad, 2018), redundancy solving (Bernstein, 1967) and error-based learning (Shadmehr et al., 2010). Traditional motor adaptation experiments do not account for the redundancy present at the various levels of the human body, from joints to muscles, to neurons. In such redundant systems, sensitivity derivatives must be known to enable transformation of task errors into appropriate motor corrections (Abdelghani et al., 2008; Hadjiosif et al., 2021). However, in de novo learning tasks, there is no prior knowledge of which motor commands affect the task and which do not. Although reinforcement learning enables learning in de novo redundant systems through exploration, it has difficulty scaling for systems with more degrees of freedom (Dal'Bello & Izawa, 2021; Sutton & Barto, 2018). Motor exploration, of the entire motor command space, may be essential in

Table 1
Phenomena studied in different papers.

Paper	Phenomena of interest			
	Role of motor exploration	Error-based learning	Learning sensitivity derivative	Redundancy
Hadjiosif et al. (2021)	✖	✔	✔	✖
Hirashima and Oya (2016)	✔	✔	✖	✔
Kawato and Gomi (1992)	✖	✔	✖	✔
Pierella et al. (2019)	✖	✔	✔	✔
Rolf and Steil (2014)	✔	✖	✖	✔
Sternad (2018)	✔	✖	✖	✔
Current paper	✔	✔	✔	✔

the learning of an appropriate sensitivity derivative, enabling the successful transformation of task errors into motor commands and thus the learning, in an error-based way, of a new controller for the task.

A recent study using an EMG control task provided evidence in line with our proposed learning mechanism (Barradas et al., 2020). In that study, it was reported that motor commands that were not encoded by muscle synergies had a significant influence on task performance under an incompatible surgery (Barradas et al., 2020), with the amount of change in the angle of the artificial force generated by each muscle being related to how easy or difficult the virtual surgery was to learn. We believe that these results are in line with the current findings, which revealed that learning under incompatible virtual surgery depended on the exploration of motor commands not encoded by the previously identified synergies. In addition, larger changes in the direction of the artificial force of the muscles would be expected to require a larger update of the sensitivity derivative, therefore leading to more difficult learning. These results further confirm our hypothesis that motor exploration is essential in the learning of a sensitivity derivative for a new skill, which then enables the learning of a new controller for the motor task.

Our proposed mechanism relies on a limitation in the amplitude of a motor correction signal represented in the brain to explain how task-null variability correlates with the estimated learning rate of the controller in certain motor tasks. A similar technique has been used in the field of machine learning to avoid the problem of exploding gradients encountered in recurrent neural networks (Pascanu et al., 2013). We believe that the likely encoding of such motor correction via the firing rate of a population of neurons (Dayan & Abbott, 2001) naturally limits the amplitude of such signals because of biological constraints, such as a limit to the firing rate of neurons. Such a limitation has been observed in traditional motor adaptation experiments in humans, where it is common for not all of the error to be corrected in a single trial. Because such experiments often do not account for the redundancy of the human body, it remains to be seen whether such a limitation is also present in task-null motor corrections.

4.2. Other works do not examine the role of motor exploration on learning a sensitivity derivative for redundant error-based learning

Our research investigates the role of motor exploration on the learning of a sensitivity derivative to be used in the training of an inverse model in redundant motor tasks. Some previous works have suggested that sensitivity derivatives are innate and known (Hirashima & Oya, 2016; Kawato & Gomi, 1992). While the learning of sensitivity derivatives has been investigated in the past (Hadjiosif et al., 2021), including in redundant contexts (Pierella et al., 2019), the role of motor exploration (Sternad, 2018) in this learning has not been extensively studied.

Approaches for training a controller that rely on exploration noise but not on an internal model of a sensitivity derivative have been proposed in previous studies (Reinhart, 2017; Rolf & Steil, 2014). These approaches consist of using the current controller to generate a motor command $\boldsymbol{u}$ to the desired task outcome $\boldsymbol{y}^*$, adding exploratory noise $\boldsymbol{\epsilon}$ to the motor command, observing its outcome $\boldsymbol{y}$ in the task, and finally using the outcome–motor command pair $(\boldsymbol{y}, \boldsymbol{u}+\boldsymbol{\epsilon})$ as a training sample for the controller, with convergence to an appropriate solution guaranteed when the entire motor command space is explored (Rolf & Steil, 2014). Although such approaches lead to the successful training of a controller, they do not replicate some features of error-based adaptation observed in human experiments. When the controller is composed of motor primitives (Thoroughman & Shadmehr, 2000), this learning approach would be expected to cause a change in the motor command generated for the noisy desired outcome $\boldsymbol{y}$, but not for the original desired outcome $\boldsymbol{y}^*$. Although generalization of movement errors across the task space has been shown in humans, it peaks at the original desired task outcome (Thoroughman & Shadmehr, 2000). We believe that our proposed framework replicates the error-based adaptation observed in humans well, while also incorporating the effect of exploration on motor learning in redundant contexts. A summary of a comparison of our work with previous works can be seen in Table 1.

4.3. Possible implementation of sensitivity derivatives in the cerebello–thalamo–cortical pathway

Regarding the case in which sensitivity derivatives are not represented in forward models, we believe that they are likely represented at the cerebello–thalamo–cortical (CTC) pathway, connecting the cerebellum to the motor cortex via the thalamus (Aumann, 2002). Previous studies suggest that the integrity of the CTC pathway is related to the capacity to adapt to movement errors (Chen et al., 2006), but is not crucial for the execution of already learned movements (Fabre-Thorpe & Levesque, 1991). In our framework, assuming that sensory prediction errors are computed in the cerebellum (Shadmehr et al., 2010), these errors might be relayed to the CTC pathway, which would transform the errors into motor correction signals, which are then relayed to the motor cortex to change the cortical synapses responsible for the generation of movement. In the execution of already learned movements, there would likely be very few sensory prediction errors. Thus, the CTC pathway would not be crucial in the generation of movements in a feedforward way. There is

also evidence that the amount of connectivity between the motor cortex and cerebellum, partially through the CTC pathway, is related to residual motor output in chronic stroke patients (Schulz et al., 2015), suggesting its role in the learning of a new controller. In cases of stroke where the death of neurons causes a change in the relationship between neuronal activation and motor task outcome, the sensitivity derivative would likely need to be learned again, similar to what we propose is a core feature of a de novo learning task. The CTC pathway has also been reported to exhibit plasticity during motor learning (Biane et al., 2016) and sensorimotor learning (Audette et al., 2019), even after the so-called critical period of motor recovery in stroke (Yu et al., 2012). These findings suggest that, if the CTC pathway has the function of sensitivity derivative during motor learning, it could be updated with training. Although additional experimental confirmation of this possibility is required, the CTC pathway is a reasonable candidate for a neural pathway in which a sensitivity derivative could be implemented in the human brain.

4.4. Implications on the acquisition of new motor skills and in rehabilitation

Our findings suggest that motor exploration is a key factor in the learning of new skills. It has been suggested that the basal ganglia, which are thought to be responsible in the regulation of movement variability (Ölveczky et al., 2005; Pekny et al., 2015), play an important role in de novo motor learning (Gutierrez-Garralda et al., 2013; Krakauer et al., 2019). During the acquisition of new motor skills, such as when learning to use a new BMI (Sadtler et al., 2014; Sussillo et al., 2016), or in settings where the relationship between neural activation and task outcome changes considerably, such as in stroke rehabilitation (Krakauer & Carmichael, 2017), it might be beneficial for participants learning the task to increase their motor variability, at least during the initial acquisition of a sensitivity derivative appropriate for the task. Although this suggestion is not new (Sternad, 2018), in the current study we tested a mechanism that explains in detail how such an increase in variability could be useful for the motor system to learn a new task. Knowledge of this motor learning mechanism could be useful in the development of new, more efficient training strategies for users of BMIs and for rehabilitation therapies for stroke patients in the future.

4.5. Limitations

The derivation of our predictions about our proposed learning system rely on the assumption that both task environment, inverse model, and forward model are linear systems. We tried to compensate for this limitation by testing our approach in nonlinear tasks, such as the arm-reaching task from Singh et al. (2016), where we also used nonlinear activation functions in the inverse models, and we also used radial basis functions for both inverse and forward models in our simulations of the EMG task from Berger et al. (2013).

Another limitation is in our representation of sensitivity derivatives as an error backpropagation through a forward model. Since so little is known about how sensitivity derivatives are implemented in the brain, we opted to represent them in a way in which they have been studied in the past (Jordan & Rumelhart, 1992; Pierella et al., 2019), even though it is still unknown whether such backpropagation operation is implemented in the brain (Lillicrap et al., 2020). In the future we plan to test other architectures for the sensitivity derivatives, such as feedforward neural networks which receive as input the task error and the motor command and output the motor corrections, which are then used in the training of the inverse models. Further studies into the possible role of the CTC pathway as a sensitivity derivative might also elucidate the best architecture to use for such function in simulations.

Another limitation in our study is in not allowing for online, feedback-driven movement corrections to be made. There is evidence that feedforward and feedback control mechanisms are learned separately in de novo tasks (Kasuga et al., 2015), which might suggest that they possess separate underlying learning processes. Variables such as the presence or not of online visual feedback have shown to differentially influence the extent to which feedforward and feedback learning processes are recruited (Batcho et al., 2016). Further studies could help elucidate the interaction between the learning of the feedforward and feedback components of motor control, and whether the learning of the feedback control mechanism also benefits from motor exploration, and if so, in what way.

5. Conclusion

We proposed a motor learning mechanism by which motor exploration is used to train a sensitivity derivative that enables the subsequent learning of a controller in a novel motor task. We analyzed the basic properties of our proposed learning mechanism with equations. The results of simulations revealed that the proposed learning mechanism was able to reproduce many of the properties observed in various motor learning tasks. These findings suggest that a similar learning mechanism might be implemented in the human brain.

Our proposed learning mechanism provides insights regarding the optimal learning strategy in de novo tasks. The results indicated that motor exploration is a major factor in the learning of a new motor skill in situations where there is no prior knowledge of which motor commands should be used in the task. Although further confirmation through additional experiments is required, the current findings suggest that motor exploration should be encouraged during the learning of a new skill. Thus, the current findings indicate a potentially valuable future direction for the development of new therapies in the context of motor rehabilitation.

CRediT authorship contribution statement

Lucas Rebelo Dal'Bello: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft preparation, Writing – review and editing, Visualization. **Jun Izawa:** Conceptualization, Methodology, Investigation, Writing – review and editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by KAKENHI (Scientific Research on Innovative Areas 19H04977 and 19H05729). LD was supported by a Japanese Government (Monbukagakusho: MEXT) Scholarship.

References

Abdelghani, M. N., Lillicrap, T. P., & Tweed, D. B. (2008). Sensitivity derivatives for flexible sensorimotor learning. *Neural Computation*, *20*(8), 2085–2111. http://dx.doi.org/10.1162/neco.2008.04-07-507.

Audette, N. J., Bernhard, S. M., Ray, A., Stewart, L. T., & Barth, A. L. (2019). Rapid plasticity of higher-order thalamocortical inputs during sensory learning. *Neuron*, *103*(2), 277–291.e4. http://dx.doi.org/10.1016/j.neuron.2019.04.037.

Aumann, T. D. (2002). Cerebello-thalamic synapses and motor adaptation. *The Cerebellum*, *1*(1), 69–77. http://dx.doi.org/10.1080/147342202753203104.

Barradas, V. R., Kutch, J. J., Kawase, T., Koike, Y., & Schweighofer, N. (2020). When 90% of the variance is not enough: Residual EMG from muscle synergy extraction influences task performance. *Journal of Neurophysiology*, *123*(6), 2180–2190. http://dx.doi.org/10.1152/jn.00472.2019.

Batcho, C. S., Gagné, M., Bouyer, L. J., Roy, J. S., & Mercier, C. (2016). Impact of online visual feedback on motor acquisition and retention when learning to reach in a force field. *Neuroscience*, *337*, 267–275. http://dx.doi.org/10.1016/j.neuroscience.2016.09.020.

Berger, D. J., Gentner, R., Edmunds, T., Pai, D. K., & D'Avella, A. (2013). Differences in adaptation rates after virtual surgeries provide direct evidence for modularity. *Journal of Neuroscience*, *33*(30), 12384–12394. http://dx.doi.org/10.1523/JNEUROSCI.0122-13.2013.

Bernstein, N. A. (1967). *The co-ordination and regulation of movements*. Pergamon Press.

Biane, J. S., Takashima, Y., Scanziani, M., Conner, J. M., & Tuszynski, M. H. (2016). Thalamocortical projections onto behaviorally relevant neurons exhibit plasticity during adult motor learning. *Neuron*, *89*(6), 1173–1179. http://dx.doi.org/10.1016/j.neuron.2016.02.001.

Chen, H., Hua, S. E., Smith, M. A., Lenz, F. A., & Shadmehr, R. (2006). Effects of human cerebellar thalamus disruption on adaptive control of reaching. *Cerebral Cortex*, *16*(10), 1462–1473. http://dx.doi.org/10.1093/cercor/bhj087.

Dal'Bello, L. R., & Izawa, J. (2021). Task-relevant and task-irrelevant variability causally shape error-based motor learning. *Neural Networks*, *142*, 583–596. http://dx.doi.org/10.1016/j.neunet.2021.07.015.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Massachusetts Institute of Technology Press.

de Boor, C. (2002). Convergence of the power sequence. In *Applied linear algebra* (pp. 111–120). http://digital.library.wisc.edu/1793/11635.

Desmurget, M., Reilly, K. T., Richard, N., Szathmari, A., Mottolese, C., & Sirigu, A. (2009). Movement intention after parietal cortex stimulation in humans. *Science*, *324*(5928), 811–813. http://dx.doi.org/10.1126/science.1169896.

Dhawale, A. K., Smith, M. A., & Ölveczky, B. P. (2017). The role of variability in motor learning. *Annual Review of Neuroscience*, *40*(1), 479–498. http://dx.doi.org/10.1146/annurev-neuro-072116-031548.

Fabre-Thorpe, M., & Levesque, F. (1991). Visuomotor relearning after brain damage crucially depends on the integrity of the ventrolateral thalamic nucleus. *Behavioral Neuroscience*, *105*(1), 176–192. http://dx.doi.org/10.1037/0735-7044.105.1.176.

Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, *233*(4771), 1416–1419. http://dx.doi.org/10.1126/science.3749885.

Gutierrez-Garralda, J. M., Moreno-Briseño, P., Boll, M.-C., Morgado-Valle, C., Campos-Romo, A., Diaz, R., & Fernandez-Ruiz, J. (2013). The effect of Parkinson's disease and huntington's disease on human visuomotor learning. *European Journal of Neuroscience*, http://dx.doi.org/10.1111/ejn.12288, n/a-n/a.

Hadjiosif, A. M., Krakauer, J. W., & Haith, A. M. (2021). Did we get sensorimotor adaptation wrong? Implicit adaptation as direct policy updating rather than forward-model-based learning. *The Journal of Neuroscience*, *41*(12), 2747–2761. http://dx.doi.org/10.1523/JNEUROSCI.2125-20.2021.

Haith, A. M., & Krakauer, J. W. (2013). Model-based and model-free mechanisms of human motor learning. In M. J. Richardson, M. A. Riley, & K. Shockley (Eds.), *Progress in motor control, Vol. 782* (pp. 1–21). New York: Springer, http://dx.doi.org/10.1007/978-1-4614-5465-6_1.

Hennig, J. A., Golub, M. D., Lund, P. J., Sadtler, P. T., Oby, E. R., Quick, K. M., Ryu, S. I., Tyler-Kabara, E. C., Batista, A. P., Yu, B. M., & Chase, S. M. (2018). Constraints on neural redundancy. *ELife*, *7*, Article e36774. http://dx.doi.org/10.7554/eLife.36774.

Hirashima, M., & Oya, T. (2016). How does the brain solve muscle redundancy? Filling the gap between optimization and muscle synergy hypotheses. *Neuroscience Research*, *104*, 80–87. http://dx.doi.org/10.1016/j.neures.2015.12.008.

Inoue, M., Uchimura, M., & Kitazawa, S. (2016). Error signals in motor cortices drive adaptation in reaching. *Neuron*, *90*(5), 1114–1126. http://dx.doi.org/10.1016/j.neuron.2016.04.029.

Izawa, J., & Shadmehr, R. (2011). Learning from sensory and reward prediction errors during motor adaptation. *PLoS Computational Biology*, *7*(3), Article e1002012. http://dx.doi.org/10.1371/journal.pcbi.1002012.

Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, *16*(3), 307–354. http://dx.doi.org/10.1207/s15516709cog1603_1.

Kalaska, J. F. (2009). From intention to action: Motor cortex and the control of reaching movements. In D. Sternad (Ed.), *Progress in motor control, Vol. 629* (pp. 139–178). Springer US, http://dx.doi.org/10.1007/978-0-387-77064-2_8.

Kasuga, S., Telgen, S., Ushiba, J., Nozaki, D., & Diedrichsen, J. (2015). Learning feedback and feedforward control in a mirror-reversed visual environment. *Journal of Neurophysiology*, *114*(4), 2187–2193. http://dx.doi.org/10.1152/jn.00096.2015.

Kawato, M., & Gomi, H. (1992). A computational model of four regions of the cerebellum based on feedback-error learning. *Biological Cybernetics*, *68*(2), 95–103. http://dx.doi.org/10.1007/BF00201431.

Krakauer, J. W., & Carmichael, S. T. (2017). *Broken movement: the neurobiology of motor recovery after stroke*. The MIT Press.

Krakauer, J. W., Hadjiosif, A. M., Xu, J., Wong, A. L., & Haith, A. M. (2019). Motor learning. *Comprehensive Physiology*, *9*(2), 613–663. http://dx.doi.org/10.1002/cphy.c170043.

Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, *7*(1), http://dx.doi.org/10.1038/ncomms13276.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*(6), 335–346. http://dx.doi.org/10.1038/s41583-020-0277-3.

Liu, X., Mosier, K. M., Mussa-Ivaldi, F. A., Casadio, M., & Scheidt, R. A. (2011). Reorganization of finger coordination patterns during adaptation to rotation and scaling of a newly learned sensorimotor transformation. *Journal of Neurophysiology*, *105*(1), 454–473. http://dx.doi.org/10.1152/jn.00247.2010.

Mosier, K. M., Scheidt, R. A., Acosta, S., & Mussa-Ivaldi, F. A. (2005). Remapping hand movements in a novel geometrical environment. *Journal of Neurophysiology*, *94*(6), 4362–4372. http://dx.doi.org/10.1152/jn.00380.2005.

Ölveczky, B. P., Andalman, A. S., & Fee, M. S. (2005). Vocal experimentation in the juvenile songbird requires a basal ganglia circuit. *PLoS Biology*, *3*(5), Article e153. http://dx.doi.org/10.1371/journal.pbio.0030153.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th international conference on international conference on machine learning - Vol. 28* (pp. III-1310-III-1318).

Pekny, S. E., Izawa, J., & Shadmehr, R. (2015). Reward-dependent modulation of movement variability. *Journal of Neuroscience*, *35*(9), 4015–4024. http://dx.doi.org/10.1523/JNEUROSCI.3244-14.2015.

Pierella, C., Casadio, M., Mussa-Ivaldi, F. A., & Solla, S. A. (2019). The dynamics of motor learning through the formation of internal models. *PLoS Computational Biology*, *15*(12), Article e1007118. http://dx.doi.org/10.1371/journal.pcbi.1007118.

Reinhart, R. F. (2017). Autonomous exploration of motor skills by skill babbling. *Autonomous Robots*, *41*(7), 1521–1537. http://dx.doi.org/10.1007/s10514-016-9613-x.

Rohde, M., Narioka, K., Steil, J. J., Klein, L. K., & Ernst, M. O. (2019). Goal-related feedback guides motor exploration and redundancy resolution in human motor skill acquisition. *PLoS Computational Biology*, *15*(3), Article e1006676. http://dx.doi.org/10.1371/journal.pcbi.1006676.

Rolf, M., & Steil, J. J. (2014). Explorative learning of inverse models: A theoretical perspective. *Neurocomputing*, *131*, 2–14. http://dx.doi.org/10.1016/j.neucom.2013.04.050.

Sadtler, P. T., Quick, K. M., Golub, M. D., Chase, S. M., Ryu, S. I., Tyler-Kabara, E. C., Yu, B. M., & Batista, A. P. (2014). Neural constraints on learning. *Nature*, *512*(7515), 423–426. http://dx.doi.org/10.1038/nature13665.

Schulz, R., Frey, B. M., Koch, P., Zimerman, M., Bönstrup, M., Feldheim, J., Timmermann, J. E., Schön, G., Cheng, B., Thomalla, G., Gerloff, C., & Hummel, F. C. (2015). Cortico-cerebellar structural connectivity is related to residual motor output in chronic stroke. *Cerebral Cortex*, Article bhv251. http://dx.doi.org/10.1093/cercor/bhv251.

Shadmehr, R., Smith, M. A., & Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annual Review of Neuroscience*, *33*(1), 89–108. http://dx.doi.org/10.1146/annurev-neuro-060909-153135.

Singh, P., Jana, S., Ghosal, A., & Murthy, A. (2016). Exploration of joint redundancy but not task space variability facilitates supervised motor learning. *Proceedings of the National Academy of Sciences*, *113*(50), 14414–14419. http://dx.doi.org/10.1073/pnas.1613383113.

Sternad, D. (2018). It's not (only) the mean that matters: Variability, noise and exploration in skill learning. *Current Opinion in Behavioral Sciences*, *20*, 183–195. http://dx.doi.org/10.1016/j.cobeha.2018.01.004.

Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., & Shenoy, K. V. (2016). Making brain–machine interfaces robust to future neural variability. *Nature Communications*, *7*(1), 13749. http://dx.doi.org/10.1038/ncomms13749.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: an introduction* (2nd ed.). The MIT Press.

Tanaka, H., Ishikawa, T., Lee, J., & Kakei, S. (2020). The cerebro-cerebellum as a locus of forward model: A review. *Frontiers in Systems Neuroscience*, *14*(19), http://dx.doi.org/10.3389/fnsys.2020.00019.

Telgen, S., Parvin, D., & Diedrichsen, J. (2014). Mirror reversal and visual rotation are learned and consolidated via separate mechanisms: Recalibrating or learning de novo? *The Journal of Neuroscience*, *34*(41), 13768–13779. http://dx.doi.org/10.1523/JNEUROSCI.5306-13.2014.

Thoroughman, K. A., & Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Nature*, *407*(6805), 742–747. http://dx.doi.org/10.1038/35037588.

Thorp, E. B., Kording, K. P., & Mussa-Ivaldi, F. A. (2017). Using noise to shape motor learning. *Journal of Neurophysiology*, *117*(2), 728–737. http://dx.doi.org/10.1152/jn.00493.2016.

Tseng, Y., Diedrichsen, J., Krakauer, J. W., Shadmehr, R., & Bastian, A. J. (2007). Sensory prediction errors drive cerebellum-dependent adaptation of reaching. *Journal of Neurophysiology*, *98*(1), 54–62. http://dx.doi.org/10.1152/jn.00266.2007.

Wu, H. G., Miyamoto, Y. R., Castro, L. N. G., Ölveczky, B. P., & Smith, M. A. (2014). Temporal structure of motor variability is dynamically regulated and predicts motor learning ability. *Nature Neuroscience*, *17*(2), 312–321. http://dx.doi.org/10.1038/nn.3616.

Yang, C. S., Cowan, N. J., & Haith, A. M. (2021). De novo learning versus adaptation of continuous control in a manual tracking task. *ELife*, *10*, Article e62578. http://dx.doi.org/10.7554/eLife.62578.

Yu, X., Chung, S., Chen, D.-Y., Wang, S., Dodd, S. J., Walters, J. R., Isaac, J. T. R., & Koretsky, A. P. (2012). Thalamocortical inputs show post-critical-period plasticity. *Neuron*, *74*(4), 731–742. http://dx.doi.org/10.1016/j.neuron.2012.04.024.