nature neuroscience

Article

https://doi.org/10.1038/s41593-023-01293-9

Schema formation in a neural population subspace underlies learning-to-learn in flexible sensorimotor problem-solving

Received: 2 September 2021

Accepted: 27 February 2023

Published online: 6 April 2023

Check for updates

Vishwa Goudar[1], Barbara Peysakhovich[2], David J. Freedman[2], Elizabeth A. Buffalo[3,4] & Xiao-Jing Wang[1]

Learning-to-learn, a progressive speedup of learning while solving a series of similar problems, represents a core process of knowledge acquisition that draws attention in both neuroscience and artificial intelligence. To investigate its underlying brain mechanism, we trained a recurrent neural network model on arbitrary sensorimotor mappings known to depend on the prefrontal cortex. The network displayed an exponential time course of accelerated learning. The neural substrate of a schema emerges within a low-dimensional subspace of population activity; its reuse in new problems facilitates learning by limiting connection weight changes. Our work highlights the weight-driven modifications of the vector field, which determines the population trajectory of a recurrent network and behavior. Such plasticity is especially important for preserving and reusing the learned schema in spite of undesirable changes of the vector field due to the transition to learning a new problem; the accumulated changes across problems account for the learning-to-learn dynamics.

In psychology, a 'schema' is an abstract mental representation deployed to interpret and respond to new experiences and to recall these experiences later from memory[1,2]. Mental schemas are thought to express knowledge garnered from past experiences[2–4]. For example, expert physicists apply relevant schemas when they categorize mechanics problems based on governing physical principles (for example, conservation of energy or Newton's second law); by contrast, novice physicists who lack these schemas resort to categories based on concrete problem cues (for example, objects in the problem or their physical configuration)[5]. What is the brain mechanism of schemas, and what makes it essential for rapid learning and abstraction?

One type of schema is called a 'learning set'. In a pioneering experiment, H. F. Harlow trained macaque monkeys on a series of stimulus–reward association problems[6]. While keeping the task structure fixed, each problem consisted of two novel stimuli that had to be correctly mapped onto rewarded versus non-rewarded, respectively. Harlow found that the monkeys progressively improved their learning efficiency over the course of a few hundred problems, until they learned new problems in one shot. He concluded that, rather than learning each problem independently of the earlier ones, the monkeys formed an abstract learning set that they deployed to learn new problems more efficiently—they were 'learning-to-learn'.

Schemas are posited to emerge as an abstraction of the commonalities across previous experiences[4,7], whose generalization to novel situations accelerates learning[8–10]. Indeed, the abstract neural representation of shared task variables has been observed across consecutively learned problems when experience on earlier problems facilitates later learning[11,12]. Furthermore, the progressive improvement in learning efficiency observed by Harlow suggests that this process of abstract representation-facilitated learning undergoes progressive refinement. The structure learning hypothesis[13] equates learning to a change in the brain's internal parameters that control behavior

[1]Center for Neural Science, New York University, New York, NY, USA. [2]Department of Neurobiology, University of Chicago, Chicago, IL, USA. [3]Department of Physiology and Biophysics, University of Washington School of Medicine, Seattle, WA, USA. [4]Washington National Primate Research Center, Seattle, WA, USA. e-mail: xjwang@nyu.edu

and posits that the progressive improvement in learning efficiency emerges with a low-dimensional task-appropriate realization of the internal parameter space. Parameter exploration within such a space is less demanding, which makes learning more efficient. Therefore, whereas schema formation emphasizes an abstraction of the task's structure, structure learning emphasizes learning how to efficiently use a schema to aid in generalization. Conceptual theory notwithstanding, how, mechanistically, a neural circuit realizes a schema and applies it to expedite learning remains to be elucidated.

In spite of tremendous progress in machine intelligence, learning-to-learn presents a major challenge in presently available artificial systems. Machine learning studies have proposed 'meta-learning' approaches wherein model parameters that promote rapid generalization to new problems are explicitly favored and sought[14,15]. However, it is not known whether such mechanisms are necessary computationally or present in the brain. Can learning-to-learn arise solely from the natural dynamics of learning? We explored this question of broad interest to brain research, cognitive science and artificial intelligence by examining the neural mechanisms of learning-to-learn in recurrent neural networks (RNNs). We chose learning of arbitrary sensorimotor associations, which is essential for flexible behavior[16], as our behavioral paradigm. Here, arbitrary mappings between sensory stimuli and motor consequents must be learned on each problem[17,18]. Macaque monkeys exhibit learning-to-learn on association problems; they learn new problems within an average of 20 trials when they are well trained[19]. Furthermore, their prefrontal cortex is causally engaged during rapid problem learning. Prefrontal neurons represent task stimuli and responses during visuomotor association trials[17,19]. Prefrontal lesions produce substantial visuomotor association learning deficits[16,20,21]. We sought to understand whether and how a sensorimotor association schema may be encoded by these prefrontal representations, how it is applied to new problems and how its usage is refined to improve learning efficiency.

We found that RNNs trained on a series of sensorimotor association problems exhibit robust learning-to-learn despite the absence of meta-learning: the number of trials to learn a problem decays exponentially with the number of previously learned problems without an explicit mechanism to accelerate learning with increasing experience. We analyzed the population activity of the RNN's units via subspace decomposition to uncover population-level latent variable representations[22,23], and we used manifold perturbations to study the causal relationship between learning efficiency and the reuse of existing population representations to learn[24]. The analyses revealed that the model develops neural correlates of the task's schema—a low-dimensional neural manifold that represents shared task variables in an abstract form across problems. Its reuse avoids the formation of representations de novo while learning problems, which accelerates learning by limiting the connection weight changes required. We introduce a novel measure relating these weight modifications to population activity changes, which we term the 'weight-driven vector field change'. This measure showed that the reused representations are not entirely invariant across problems. Instead, mapping new stimuli can modify the reused representations in undesirable ways. Connection weight changes are primarily recruited to prevent such modifications. Moreover, the weight changes in early problems improve the invariance of the reused representations, limiting the degree to which they would be modified in the future, which further accelerates learning. The accumulation of such improvements over a series of problems supports structure learning and promotes learning-to-learn.

Results

Learning-to-learn in trained neural networks without meta-learning

We evaluated whether an RNN model could demonstrate learning-to-learn on delayed sensorimotor association problems. In each problem, the model learned a unique mapping between a pair of sensory stimuli (for example, images) and a pair of motor responses (Fig. 1a). Each trial began with a 0.5-second sample epoch, when a sensory stimulus was presented together with a fixation cue, and the model was required to maintain fixation. A 1-second delay epoch followed, when the model had to continue fixation in the absence of the sample stimulus. The trial concluded with a 0.5-second choice epoch signalled by removal of the fixation cue, when the model had to report its choice of the appropriate motor response. The two sample stimuli in each problem were randomly generated. The model was composed of a population of recurrently (or laterally) connected firing rate units that received 11 inputs, one signaling fixation and ten signaling features of a sample stimulus (Fig. 1b). Such stimulus representations are consistent with the finding that visual objects are represented in the monkey inferotemporal cortex by a feature-based topographic map[25]. The model is also consistent with lesion studies demonstrating the causal involvement of inferotemporal–prefrontal connections in visuomotor learning and retention[20,26]. Response choices were read out from the population's activity by three output units that represented fixation, motor response 1 or motor response 2.

The model was trained on a problem one trial at a time. Its parameters were adjusted at the end of each trial to minimize the errors in its output responses, until the output responses achieved criterion accuracy (Methods). The model was then transitioned to a new problem (Fig. 1c). Crucially, training was performed without an explicit meta-learning objective. A network trained on a series of these problems demonstrated learning-to-learn (Fig. 1d). The network required a few thousand trials to learn the first problem, which was expected because it was initialized with random connection weights. By contrast, solving the second problem took a few hundred trials. Thereafter, the trials to criterion progressively decreased over the next few hundred problems, plateauing at an average of 20 trials per problem. This decrease was well fit by a decaying exponential function, which closely matched a 30-problem moving average of the network's trials to criterion. This performance is commensurate with learning-to-learn in macaque monkeys, which exhibit an exponential decrease in their trials to criterion when trained on a series of association problems (Peysakhovich et al., unpublished), and demonstrate learning within 15–20 trials when well trained[19]. The fit's parameters quantify the network's learning-to-learn performance: the time constant measures how quickly it produces learning-to-learn, and the learning efficiency asymptote measures its trials to criterion plateau. Although naive monkeys undergo behavioral shaping on the desired response set before they are introduced to the task, a naive network's learning efficiency on the first problem reflects learning both to generate basic responses and the specifics of the problem. To avoid this confound related to learning the response set, we quantified the network's learning-to-learn performance starting with the second problem.

We tested the robustness of these results by similarly training 30 independently initialized networks. Across these networks, the learning-to-learn time constants and asymptotes were limited to a narrow range (Fig. 1e; time constant: 47.52 ± 26.22 (mean ± s.d.); asymptote: 21.33 ± 3.85). We also found that the model's learning speed on a problem depends on the perceptual similarity between its sample stimulus pair and that of the previously learned problem (Fig. 1f), with higher similarity producing faster learning. We further tested the model over a range of hyperparameter settings (f-I transfer functions, learning rates, weight and firing rate regularization levels) and observed robust learning-to-learn across all conditions (Supplementary Fig. 1). In addition, we found that the model was faster at re-learning problems after subsequently learning several new problems (Supplementary Fig. 2), suggesting that it retains a memory of previously learned problems. Taken together, these results demonstrate that networks trained on a series of delayed sensorimotor association problems robustly exhibit learning-to-learn, despite the absence of an explicit meta-learning objective.

https://doi.org/10.1038/s41593-023-01293-9

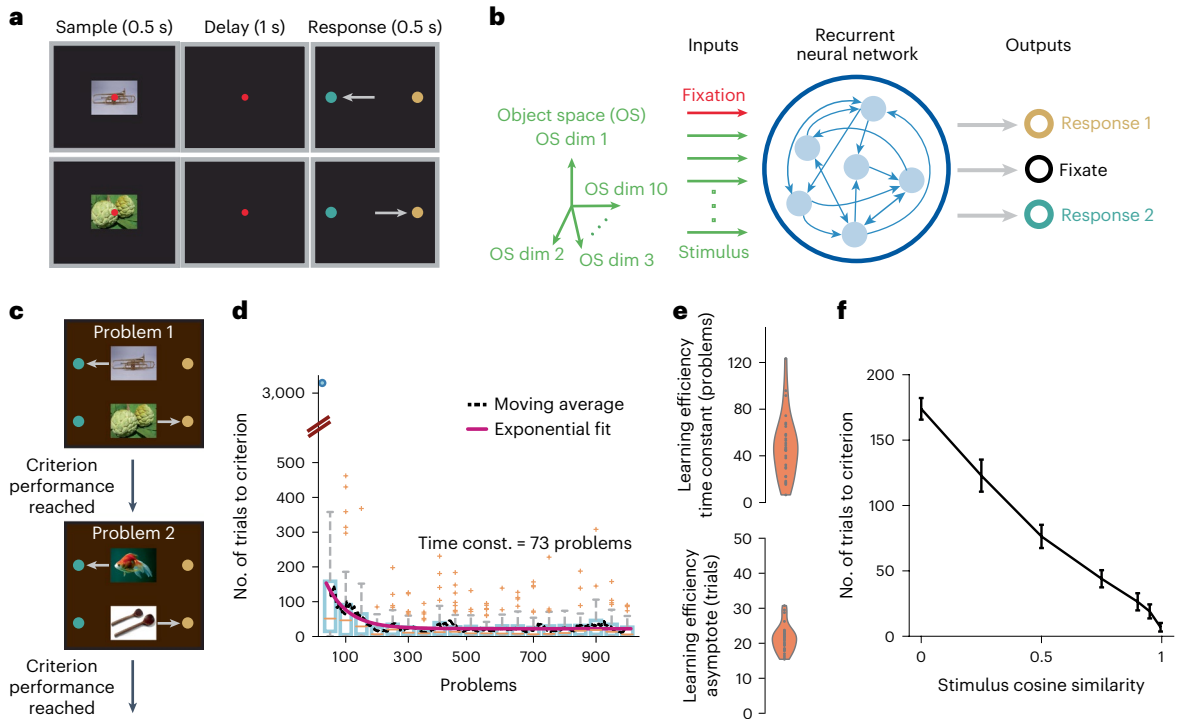

Fig. 1 | RNNs trained on delayed sensorimotor association problems exhibit learning-to-learn. a, Structure of an example delayed sensorimotor association problem. The model must learn to associate two sensory stimuli (for example, images) with corresponding motor responses (for example, a saccade). Target are colored to emphasize the distinction between response choices, not to indicate that the response targets are colored. **b**, RNN model is composed of recurrently connected rate units that receive a fixation signal and features of the sample sensory stimulus as inputs. It reports its response choices via output units corresponding to fixation, motor response choice 1 (brown) or motor response choice 2 (teal). **c**, The model is trained on a series of sensorimotor association problems, each with a randomly chosen sample stimulus pair. It is transitioned to a new problem upon reaching criterion performance on the current problem. **d**, A network's learning efficiency, measured as the number of trials to criterion performance, over 1,000 consecutively learned problems. Box plots summarize the learning efficiency in groups of 50 consecutive problems (center line: median; box bottom/top edge: 25th/75th percentiles; whiskers: minimum/maximum within 1.5× the interquartile range from box edge; +: outliers). The number of trials to criterion on a problem decreases with the number of previously learned problems. This is characterized by a decaying exponential function that demonstrates the model's ability to produce learning-to-learn. **e**, Thirty RNNs with different initial conditions exhibit learning-to-learn, as indicated by their learning-to-learn time constants (top) and asymptotes (bottom). **f**, Learning efficiency on the third problem as a function of the cosine similarity of its sample sensory stimulus pair to the previously learned problem (problem 2). Trials to criterion are averaged over 50 independently chosen stimulus pairs for each similarity value and presented as the mean and standard error (error bars) of this average across ten networks with different initial conditions. Time const., time constants.

Abstracted neural manifold governs the task's schema

The activity of a population of *N* recurrently connected units co-evolves during a trial, forming a trajectory in *N*-dimensional population state space (Fig. 2a, top). When a problem is learned, the network responds to each sample stimulus with a trajectory that appropriately subserves stimulus integration, decision-making, working memory maintenance and fixation/response choice. We demixed[27] (Methods) trajectories from consecutively learned problems to identify shared neural representations that support these computations. This procedure decomposed the trajectories into components embedded within two non-overlapping subspaces of the state space (Fig. 2a, middle). Decision representations embedded within the 'decision subspace' revealed similarities between trajectories that shared their response choice; stimulus representations embedded within the 'stimulus subspace' varied in a problem-dependent and a sample stimulus-dependent manner. We further decomposed the two decision representations in each problem into a mean decision representation, with the mean taken over both decision representations (Fig. 2a, bottom left) and residual decision representations given by subtracting out this mean from each decision representation (Fig. 2a, bottom right).

Decomposing the trajectories from the first 50 consecutively learned problems in this manner revealed a low-dimensional shared decision subspace (2.36 ± 0.18 dimensions across ten networks), whose constituent decision representations explained most of the variance in population activity across problems (88.54% ± 3.16% across ten networks). Furthermore, the mean decision representations lay close to each other in state space, forming a shared manifold across problems (Fig. 2b, left). The residual decision representations consistently encoded the decision and choice of either response across problems, thus forming a shared manifold for each decision (Fig. 2b, center). The persistence of a low-dimensional shared manifold, which explains most of the population's variance across problems, demonstrates a strong abstraction of the shared task variables that it encodes. The model retains and reuses this manifold across problems, despite changes in the stimulus set and the weight change-induced change in network dynamics that transpires while learning. Moreover, population activity changes during learning are largely determined by changes in these shared representations (Supplementary Fig. 3). In contrast, the stimulus representations (Fig. 2b, right) were higher dimensional (7.98 ± 1.48 dimensions across ten networks) but explained a small proportion of the population variance. Interestingly, the distribution of neural activity in state space at the beginning and end of problem learning closely resemble each other (Supplementary Fig. 4). These results demonstrate that the model even reuses pre-established representations when responding to novel sample stimuli and learning their mappings.

https://doi.org/10.1038/s41593-023-01293-9

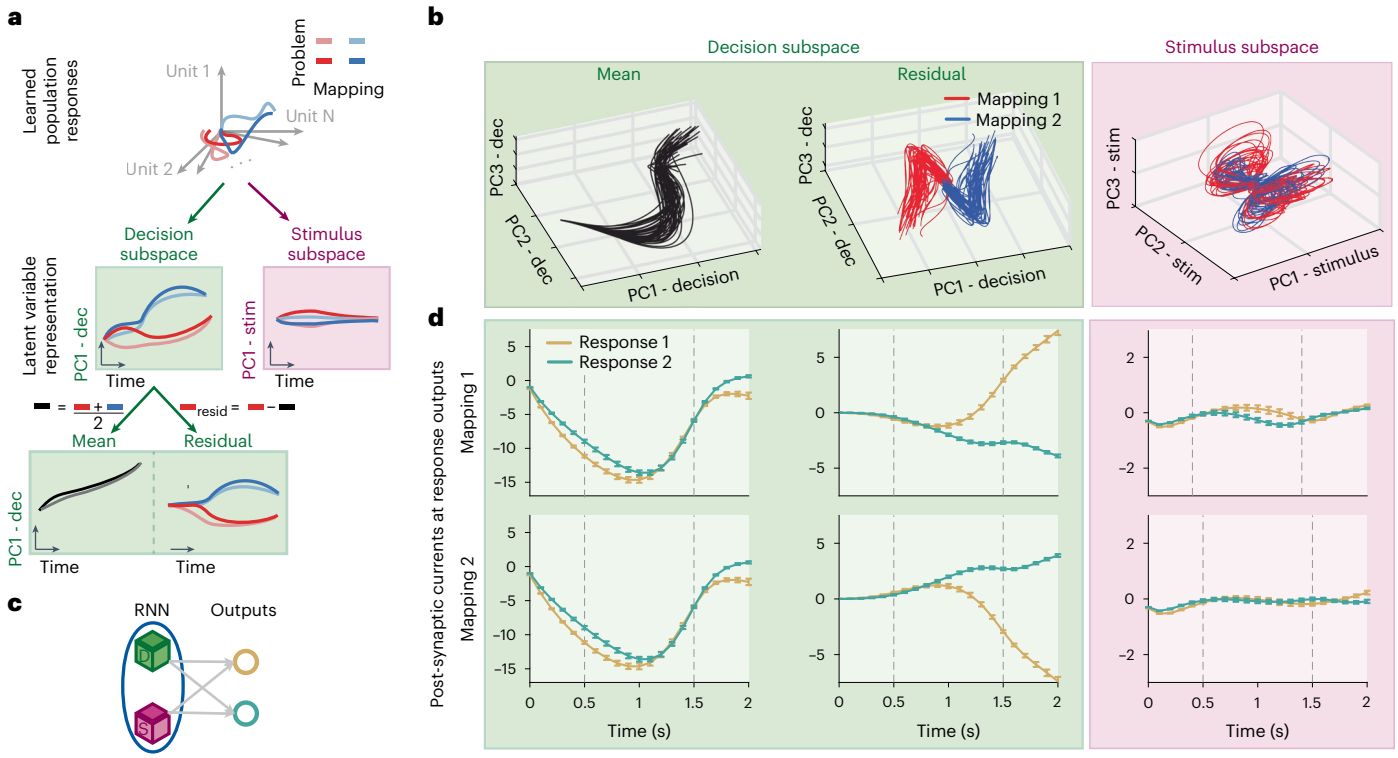

Fig. 2 | Neural representations of decision and choice are shared across problems. a, Schematic of the demixing procedure that identifies shared versus problem-dependent components of the neural representations. Population trajectories for the two mappings in 50 consecutively learned problems (illustrated for two problems, for clarity) are decomposed into components within a decision subspace, which are shared across problems by trajectories that correspond to the same response choice, and problem-dependent components embedded in a stimulus subspace. The shared decision representations are further decomposed into their mean and residual components for each problem. **b**, Decomposed representations for problems 1–50, presented along the first three principal components of their respective subspaces. **c**, Schematic illustrating that the component representations collectively drive the response choice outputs. **d**, The net current from the mean (left) and residual (center) decision representations and the stimulus representations (right) to response 1 (brown) and response 2 (teal) outputs in mapping 1 (top) and mapping 2 (bottom) trials. The mean decision components inhibit motor responses during the sample and delay epochs, and the residual decision components drive the correct response while inhibiting the incorrect one. Dashed vertical lines indicate the end of the sample and delay epochs. Plots show mean of the net currents across the 50 problems, and error bars indicate their standard errors. PC, principal component; resid, residual; stim, stimulus; dec, decision.

Next, we examined the relative contribution of these components to the output responses by measuring the net current from each component to the choice outputs (Fig. 2c). During trials where response 1 was chosen (mapping 1 trials), residual decision representations excited the response 1 output unit and inhibited the response 2 output unit, particularly within the choice epoch (Fig. 2d, center). During mapping 2 trials, these representations had the opposite effect. In contrast, the mean decision representations inhibited both response choices throughout the sample and delay epochs but not the choice epoch (Fig. 2d, left). This prevented premature choice initiation during the delay epoch (Fig. 2d, center). The contribution of stimulus representations to response selection was negligible throughout the trial (Fig. 2d, right). Quantitatively similar results were obtained for all consecutively learned 50-problem groups in all the networks that we tested. These results demonstrate that the decision manifold constitutes the neural correlates of the task's schema—it represents the shared temporal (mean decision) and two-alternative (residual decision) structure of the task in an abstract form and, thereby, reflects knowledge abstracted from past experiences.

Schema manifold scaffolds representations that facilitate learning

We have shown that the schematic decision manifold is reused by, or 'scaffolds'[28–30], the learned representations in subsequent problems. This reuse is accompanied by a stark improvement in learning efficiency between the first problem and subsequent ones (Fig. 1d). To establish whether reuse of the decision manifold causally improves learning efficiency, we compared the learning in networks that were barred from reusing it to control output responses in new problems, with networks that were allowed to do so. This method has been applied in brain–computer interface (BCI) studies to establish a causal link between monkeys' ability to rapidly adapt to BCI readout perturbations and their reuse of existing motor cortical representations to modulate the perturbed readouts[24].

In our model, this intervention relies on the concept of a 'readout subspace'. Population activity modulates an output unit's response, only when the sum of the excitatory and inhibitory post-synaptic currents it produces at the unit is non-zero (output-potent activity[31]). Therefore, the output connection weights, which mediate these currents, define a readout subspace of population state space that constrains the set of population activity levels which can modulate output responses. Our observation that population representations within the decision subspace predominantly modulate output responses implies that the decision and readout subspaces strongly overlap. Eliminating this overlap should impair the effectiveness of the pre-existing decision manifold in scaffolding newly learned trajectories and force the development of new decision representations to modulate the output responses. The observation of a concurrent learning deficit would causally link the representational scaffold to accelerated learning. For this causal intervention and its controls, we first trained a naive network

https://doi.org/10.1038/s41593-023-01293-9

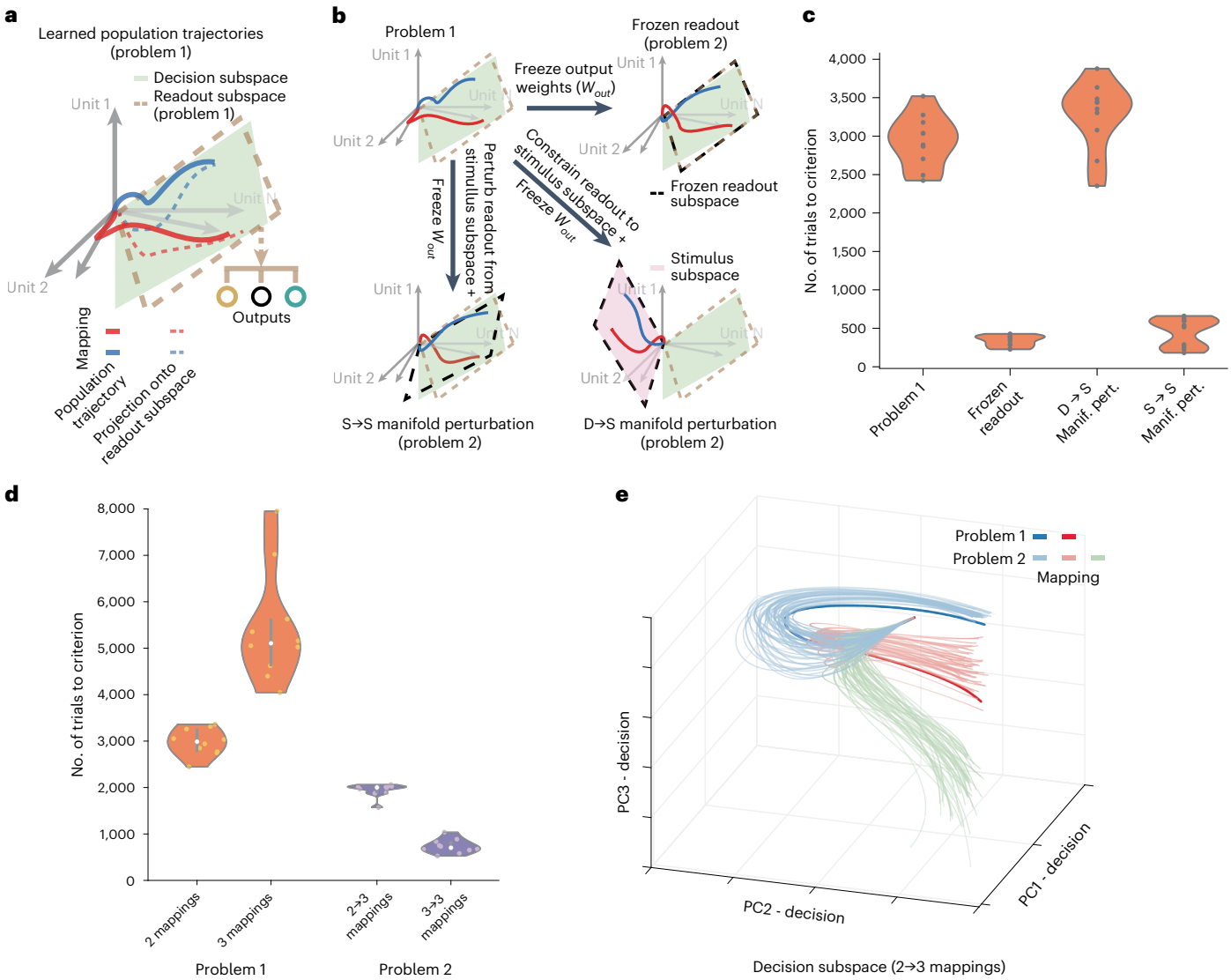

Fig. 3 | Manifold perturbations reveal that reusing the schematic decision manifold facilitates learning. a, Output responses are readout from a subspace of population state space spanned by the network's output weights. Its overlap with the decision subspace enables the control of output responses by the decision representations. **b**, Manifold perturbations to assess the role of decision manifold reuse in learning. A network is trained on its first problem to establish the decision and readout subspaces (top left). It is trained on a second problem (i) while its output weights are frozen (frozen readout, top right) (ii) after perturbing and freezing its output weights such that the readout and decision subspace overlap is eliminated (D→S manifold perturbation, bottom right) or (iii) after perturbing and freezing its output weights such that the readout and stimulus subspace overlap is altered (S→S manifold perturbation, bottom left). **c**, Average second problem learning efficiency in each of the three conditions, compared to the first problem learning efficiency. **d**, Prior knowledge transfer from a two-mapping to a three-mapping problem facilitates learning. Plot compares learning efficiency on a first problem comprising two or three mappings with the average learning efficiency on a second problem. The latter is a three-mapping problem and is preceded by either a two-mapping (2→3 mappings) or a three-mapping (3→3 mappings) problem. **e**, Learned representations for the second problem (2→3 mapping condition) in the first three principal components of the decision subspace (light) and the decision representations for the first problem projected into the same subspace (dark). Second problem decision representations are shown for 50 independently chosen stimulus sets. Trials to criterion on the second problem is averaged over 50 independently chosen random perturbations (**c**) / stimulus sets (**d**) and presented as the distribution of these averages across ten networks with different initial conditions. Box plots within violins in **d** summarize these results (center circle: median; box bottom/top edge: 25th/75th percentiles). PC, principal component; Manif. pert., manifold perturbations.

on a single problem to let it develop overlapping readout and decision subspaces (Fig. 3a).

In the frozen readout condition, we then trained the network on its second problem while freezing (or preventing changes to) the output weights (Fig. 3b, top right). Such networks exhibited a substantial improvement in learning efficiency from the first problem to the second (Fig. 3c). Thus, freezing the output weights does not adversely affect learning. In the stimulus-to-stimulus (S→S) manifold perturbation condition, we perturbed the output weights to alter the overlap between the readout and stimulus subspaces but not between the readout and decision subspaces (Fig. 3b, bottom left, and Methods). Then, we trained the network on its second problem with frozen output weights to prevent re-alignment of the readout and stimulus subspaces during training. Again, we found a substantial speedup in learning from the first problem to the second (Fig. 3c).

Finally, in the decision-to-stimulus (D→S) manifold perturbation condition, we perturbed the output weights to eliminate all overlap between the readout and decision subspaces (Fig. 3b, bottom right). We then trained the network on its second problem with frozen output weights. This compels the formation of new decision representations

within the original stimulus subspace. In contrast to the frozen readouts and S→S manifold perturbation conditions, such networks were strongly impaired at learning—they learned as slowly as naive networks learning their first problem (Fig. 3c). Collectively, these results demonstrate that impeding the reuse of the decision manifold adversely affects learning performance.

We also tested whether the transfer of prior knowledge facilitates learning of problems with altered but overlapping task structure. To do so, we trained a naive network on a single problem comprising two mappings, as in Fig. 3a. Next, we trained it on a problem comprising three mappings (that is, three sensory stimuli mapped to three motor responses). Here, too, we observed a substantial facilitation of learning performance compared to a naive network (Fig. 3d), accompanied by the reuse of the decision manifold from the two-mapping problem to learn the three-mapping problem (Fig. 3e). Taken together, these results confirm that the schematic decision manifold forms a representational scaffold that facilitates the transfer of prior knowledge regarding the task's structure to new problems and, thus, expedites learning.

Distinct roles of representation reuse and plasticity in learning

We have shown that representational reuse improves learning efficiency. However, learning produces large population activity changes to mediate the necessary output response corrections (Supplementary Fig. 7b). How does the emergence of these large changes benefit from the reuse? And how do its contributions compare to those of the plasticity-induced connection weight changes? To answer these questions, we analyzed the activity changes between the beginning and end of a problem. The population responds to a novel sample stimulus with a 'pre-learning' trajectory in state space (Fig. 4a, right, blue curve). This trajectory evolves through time via temporal integration of input and population activity mediated by input and recurrent connection weights, respectively (Eq. (2)). The resulting advance in population activity from r'_{t-1} to r'_t (Fig. 4a, left) during the brief time interval from $t-1$ to t is represented in state space by a vector originating at r'_{t-1} (Fig. 4a, right). The direction and magnitude of advance is state dependent—it depends on the activity levels of the population's units (that is, the population state) at time $t-1$. The temporal sequence of these vectors guides the evolution of population activity between the initial (r'_0) and final (r'_T) states (Fig. 4a, right, blue arrows along blue curve). These state-dependent vectors constitute a 'vector field'[32,33] that spans the entire state space and describes the network's dynamics (Fig. 4a, right, blue arrows tiling the space).

After a problem is learned, the population activity traverses a 'learned' trajectory (Fig. 4b, right, purple curve) comprising learned population states. Because the connection weights after learning are a sum of the pre-learning weights and plasticity-induced weight changes, the learned trajectory is governed by the sum of the pre-learning vector field and the change in this field due to the weight changes. Consequently, so is the change in population activity. The change in population activity from a pre-learning state (r'_T) to a learned state $(\mathbf{r}_t)$ at time t, $\mathbf{z}_t$, is represented in state space by a vector from the former to the latter (Fig. 4c, solid gray arrows). It emerges from an accumulation of activity change increments throughout the trial (Fig. 4c, green arrow). The incremental change in population activity $(\Delta\mathbf{z}_{t+1})$ between t and $t+1$ derives from the pre-learning vector field (that is, the reuse of existing representations) and the plasticity-induced change in the vector field.

Setting aside the effect of weight changes for a moment, consider the network's pre-learning vector field at the learned and pre-learning states. Due to its state dependence, the vector field may advance population activity differently from one state versus from the other. In this event, the activity difference between the pre-learning and learned states will change between times t $(\mathbf{z}_t)$ and $t+1$ $(\mathbf{z}_{t+1})$. In state space, the vector difference (Fig. 4d, left, pink arrow) between the pre-learning vector field at the two states (blue arrows) characterizes this change and is referred to as the 'state-driven vector field change' (or state-driven VFC, referred to in the Methods as $\Delta\mathbf{Field}_{s,t+1}$- Eq. (6)). The state-driven VFC depends solely on the pre-learning vector field (that is, on reused representations).

The connection weight changes alter the net post-synaptic currents into the population. This alters how its activity advances over time (Fig. 4b, left). In state space, this translates to a VFC all along the learned trajectory (Fig. 4b, right, orange arrows), including at time t (Fig. 4d, center), and it is referred to as the 'weight-driven vector field change' (or weight-driven VFC, referred to in the Methods as $\Delta\mathbf{Field}_{w,t+1}$- Eq. (7)). The sum of these two types of VFC (weight-driven and state-driven VFCs) produces the incremental change in population activity $(\Delta\mathbf{z}_{t+1})$ between t and $t+1$ (Fig. 4d, right, and Eqs. (4 and 5)).

Measurements revealed a substantial difference between the magnitudes of activity changes ($\mathbf{z}_t$; Supplementary Fig. 7b) and activity change increments ($\Delta\mathbf{z}_t$; Fig. 5b)—large activity changes emerge from an accumulation of relatively small change increments generated throughout the trial. We further assessed the relative contribution of the weight-driven and state-driven VFCs to the activity change increments by decomposing them (Fig. 5a and Methods) into their components in the direction of the activity change increments ($\Delta\mathbf{z}_{\parallel}$ - parallel component) and orthogonal to them ($\Delta\mathbf{z}_{\perp}$ - orthogonal component).

The state-driven VFC's parallel component is much larger in magnitude than the weight-driven VFC's parallel component (Fig. 5b, green bars). Therefore, the network's pre-learning vector field, which governs the state-driven VFC, is primarily responsible for the population activity changes. Furthermore, these parallel components are low dimensional not only in individual problems but also across a group of problems (Fig. 5c). This is consistent with the structure learning hypothesis[13], wherein efficient learning relies on changing behavior via changes within a low-dimensional internal parameter space of the brain. Our results suggest that this parameter space corresponds to a low-dimensional subspace of neural population activity, which constrains how population activity and behavior change while learning a problem.

The weight-driven VFC's orthogonal component is much larger in magnitude than its parallel component. Furthermore, it is equal in magnitude but opposite in direction to its state-driven counterpart and, therefore, nullifies it (Fig. 5b, pink bars). These orthogonal components are also low dimensional on individual problems but high dimensional across a group of problems (Fig. 5c). Moreover, they largely span directions along which the existing representations do not typically co-vary (Supplementary Fig. 8a). These results imply that novel sample stimuli interact with the existing representations when mapped onto them, in a manner that elicits uncharacteristic population responses. That is, the existing representations can be sensitive (that is, not entirely invariant) to the sample stimuli that are mapped onto them. The weight-driven VFC emerges primarily to impede such interactions and, thereby, prevent changes to the existing representations.

To summarize, our analysis of the population activity changes between the start and end of problem learning revealed that (1) large changes emerge over the trial time course from the accumulation of a sequence of small local changes along the learned trajectory; (2) these changes are low dimensional and stem primarily from reusing the network's pre-learning vector field to shape the learned trajectory, thus elucidating the relative contribution of representational reuse to learning; and (3) the pre-existing representations are not entirely invariant to having novel sample stimuli mapped onto them and can undergo uncharacteristic modifications in the process. Connection weight changes emerge largely to prevent such modifications.

Recurrent weight change magnitude determines learning efficiency

Next, we examined why learning efficiency is enhanced by representational reuse, by exploring how learning efficiency is impacted

https://doi.org/10.1038/s41593-023-01293-9

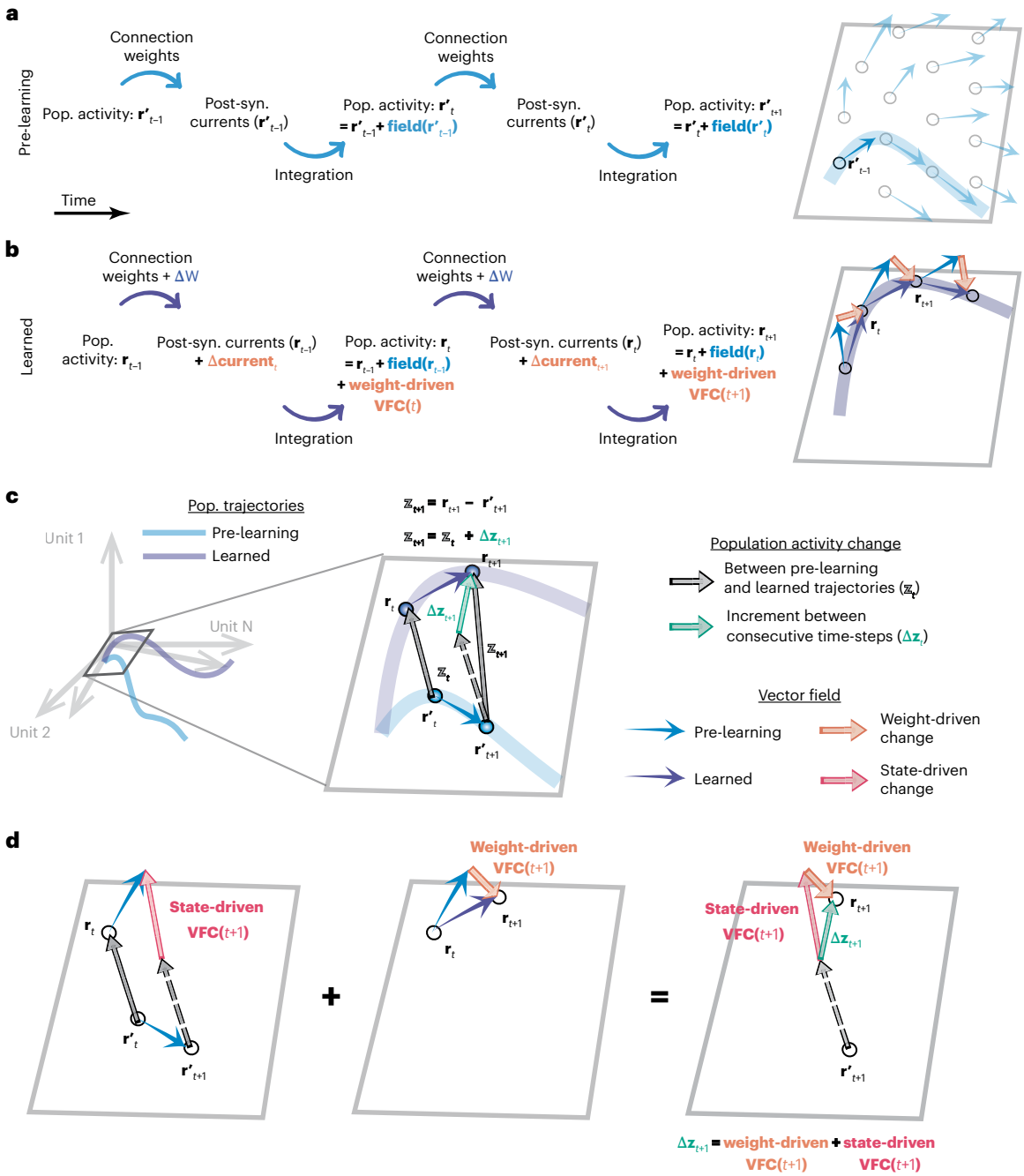

Fig. 4 | Learned trajectories emerge from VFCs. a,**b**, The temporal evolution of population activity at the start (pre-learning, **a**) and end (learned, **b**) of a problem, illustrated in population state space on the right. **a**, The activity advances due to the integration of net post-synaptic currents, which depend on the activity levels (or state) of the network and input units and their efferent connection weights (left). This population-state-dependent advance determines a vector field that tiles state space (right, blue arrows) and guides the evolution of the population trajectory (right, blue curve). **b**, Plasticity-induced connection weight changes (*ΔW*) alter the post-synaptic currents (*ΔCurrent*), thereby altering the advance in population activity (left). The effect of this weight-driven VFC is a continual series of modifications to the vector field (right, orange arrows) that determines the evolution of the learned population trajectory (right, purple curve). **c**, The divergence of the learned trajectory from the pre-learning trajectory (z_{t+1}, right, solid gray arrow) emerges from an accumulation of activity change increments throughout the trial (Δz_{t+1}, right, green arrow). **d**, Each increment is the sum of the state-driven and weight-driven VFCs (left and center, pink and orange arrows, respectively). The state-driven VFC is a result of state-dependent differences in the pre-learning vector field, specifically between learned and pre-learning population states (left, blue arrows at r_t and r'_t, respectively). Dashed gray arrows in **c** and **d** represent a displaced version of the vector $\mathbf{z}_{t+1}$ to help illustrate vector differences. Pop., population; Post-syn., post-synaptic.

by the connection weight changes. In Supplementary Note 1.1.1 and Supplementary Fig. 5, we show that the model learns via recurrent rather than input weight changes. Athough recurrent and input weight changes independently contribute to the weight-driven VFC (Eq. (7)), in the model the weight-driven VFC is determined by recurrent weight changes, as this is more efficient. Moreover, the magnitude of recurrent weight changes in a problem explains the number of trials expended in learning it (Fig. 6a). This is consistent with analytical bounds relating the magnitude of connection weight changes and sample efficiency in deep neural networks[34,35].

https://doi.org/10.1038/s41593-023-01293-9

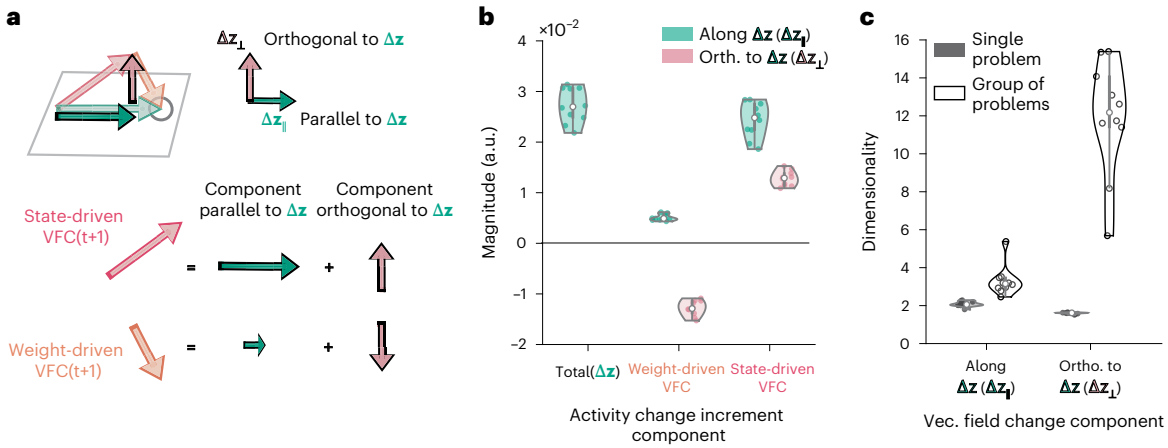

Fig. 5 | Weight-driven and state-driven VFCs differentially contribute to population activity change. a,The state-driven and weight-driven VFCs are decomposed into components along (or that contribute to) the activity change increment (green arrows) and components orthogonal to it (or that cancel out, pink arrows). **b**, Magnitude (L^2-norm) of the population activity change increments and projections of its VFC constituents (weight-driven and state-driven VFCs) in the directions along and orthogonal to the population activity change increments. Measurements shown are the temporal mean of the magnitudes over the trial duration, averaged over both mappings of problems 2–51. **c**, Dimensionality of the VFC components on single problems (averaged over problems 2–51) and for a group of 50 problems (problems 2–51). Plots represent distributions over ten networks with different initial conditions. Box plots within violins in **b** and **c** summarize these distributions (center circle: median; box bottom/top edge: 25th/75th percentiles; whiskers: minimum/maximum values). a.u., arbitrary units; Orth., orthogonal; Vec., vector.

In light of this observation and the exponential decrease in the trials to criterion across problems, we hypothesized that the magnitude of recurrent weight changes should also decrease exponentially across problems. We further posited that the magnitudes of the post-synaptic current changes and the weight-driven VFC should also decrease exponentially, because these quantities are directly related to the recurrent weight change magnitude. Figure 6b confirms that the magnitude of these three quantities decreases exponentially as a function of the number of learned problems. Therefore, the progressive improvement in the model's learning efficiency is explained by a similar decrease in the magnitudes of the recurrent weight changes and weight-driven VFC required to learn problems.

We can now explain why representational reuse markedly improves learning efficiency (Fig. 3). D→S manifold perturbations compel the development of new representations that re-encode the task's structure beyond the original decision subspace—in state space, the structure and location of these target trajectories are constrained by the arbitrarily altered output weights (Fig. 3b, bottom right). However, the vector field along such an arbitrarily constrained target trajectory is likely misaligned relative to the vector field required to support it (Supplementary Fig. 6a, right, purple versus blue arrows along the learned trajectory). Consequently, it is unlikely to roughly advance population activity along the target trajectory, as it does in unperturbed networks (Supplementary Fig. 6a, left). Measurements comparing the magnitude of the weight-driven VFC in unperturbed and perturbed networks confirms that the vector field in perturbed networks undergoes drastic re-organization in comparison to unperturbed networks (Supplementary Fig. 6b, right), so that they may support new decision representations (Supplementary Fig. 6a, large orange arrows). This explains the learning impairment after D→S manifold perturbations and demonstrates the merits of learning via representational reuse—this reuse of existing representations limits the requisite weight changes (Supplementary Fig. 6b, left) and, thereby, improves learning efficiency.

In Supplementary Note 1.1.2, we explore the reciprocal interactions between stimulus and decision representations during trial performance and learning. The analysis reveals a second form of representational scaffolding by the decision representations, wherein pre-synaptic population activity in the decision rather than the stimulus subspace modulates the weight-driven VFC (Fig. 6c and Supplementary Fig. 7d).

Accumulation of weight changes progressively speeds up learning

In agreement with Harlow's learning-to-learn experiments, our model exhibits a progressive improvement in learning efficiency spanning a few hundred problems (Fig. 1). This is explained by a progressive decrease in the magnitudes of the weight changes and weight-driven VFC per problem (Fig. 6a,b). Because the weight-driven VFC prevents distortions to existing representations during learning (Fig. 5b), a progressive decrease in its magnitude amounts to a progressive improvement in the invariance of the existing representations to learning novel mappings. However, the source of this improvement is as of yet undetermined: what causes it in the absence of an explicit meta-learning mechanism? We hypothesized that the accumulation of weight changes over earlier problems facilitates learning in future problems. That is, weight changes elicited while learning problems $p-k$ (for $1 \le k \le p-2$) cumulatively alter the vector field such that they suppress the weight-driven VFC required to learn problem p (Supplementary Fig. 9a, top, and Methods). More generally, as problems are learned, their respective weight-driven VFCs accumulate to produce a cumulative VFC, which suppresses the weight-driven VFC required to learn subsequent problems. This progressively improves representational invariance and, thereby, accelerates learning.

To test this hypothesis, for each problem p, we measured the magnitudes of its weight-driven VFC plus the cumulative VFC along its learned trajectory due to the accumulation of weight changes over the sequence of problems that precede it, from problem $p-1$ (relative problem −1) to problem 2 (relative problem $2-p$). Figure 7a summarizes these measurements across many problems p grouped by their learning-to-learn stage—that is, the number of problems they are preceded by. Here, we focused on the magnitude along each problem's orthogonal weight-driven VFC component ($\Delta \mathbf{z}_\perp$) because it dominates the total weight-driven VFC in problems at each learning-to-learn stage (Supplementary Fig. 8b). The results show that, at each stage, learning earlier problems cumulatively suppresses the weight-driven VFC required in subsequent problems. We further found that this is predominantly due to an accumulation of recurrent weight changes (Supplementary Fig. 8c). These findings confirmed our hypothesis: the accumulation of weight changes over problems progressively improves representational invariance and, therefore, learning

https://doi.org/10.1038/s41593-023-01293-9

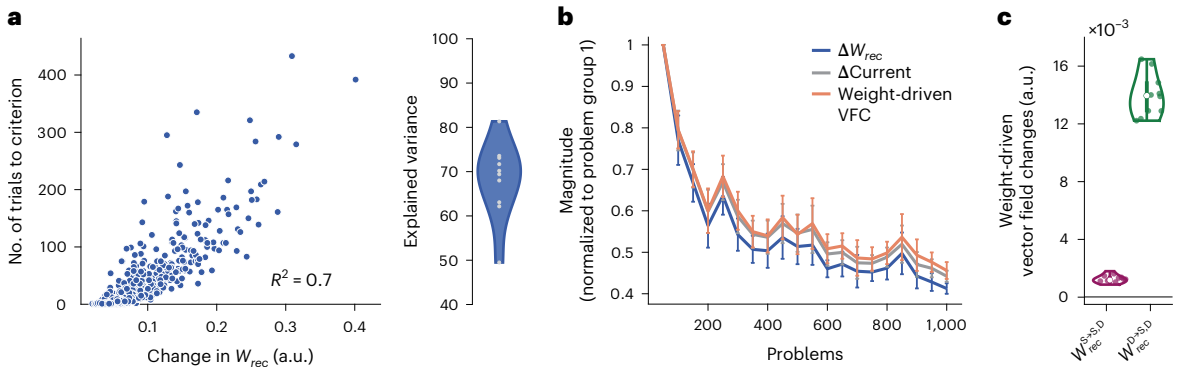

Fig. 6 | The magnitude of recurrent weight changes explains both the magnitude of the weight-driven VFC and the number of trials to learn a problem. a, The magnitude of the plasticity-induced recurrent connection weight changes explains most of the variance in the number of trials to learn problems (left). This relationship was robustly observed across ten networks with different initial conditions (right). **b**, The magnitude of recurrent weight (blue), post-synaptic current (gray) and weight-driven vector field (orange) changes, averaged in groups of 50 non-overlapping and consecutively learned problems. Each quantity was normalized by its corresponding value for the first problem group. All quantities decrease exponentially with the number of previously learned problems. **c**, Approximate contribution of pre-synaptic population activity in the stimulus versus decision subspace to the weight-driven VFC, averaged over problems 2–51. The magnitudes (L^2-norm) of the change in the post-synaptic currents and vector field represent their temporal mean over the entire trial duration, averaged over both mappings in each problem. The magnitude of recurrent weight changes was measured by their Frobenius norm. Plot **b** (plot **c**) reflects mean values (the distribution) over ten networks with different initial conditions. Error bars in **b** indicate standard errors. Box plots within violins in **c** summarize results across the ten networks (center circle: median; box bottom/top edge: 25th/75th percentiles; whiskers: minimum/maximum values). a.u., arbitrary units.

efficiency. Moreover, they imply that the cumulative change along the orthogonal weight-driven VFC component of problems imposes a learning efficiency bottleneck.

Figure 7a demonstrates that the weight-driven VFC in a problem depends on its net suppression by the preceding problems—that is, the sum of the suppressive cumulative VFC contributions (and enhancing cumulative VFC contributions, when they increase the requisite weight-driven VFC) by the weight changes in each preceding problem going back to problem 2 (Supplementary Fig. 9b, left, and Methods). A larger net suppression produces a smaller weight-driven VFC. Because the weight-driven VFC decays exponentially with the number of preceding problems (Fig. 6b), we posited that the net suppression must similarly increase with it. Measurements of the net suppression along the orthogonal and parallel weight-driven VFC components confirmed this (Fig. 7b). The net suppression mirrors the exponential decay in the weight-driven VFC (Methods)—it rapidly increases across problems at the early stages of learning-to-learn, which produces a rapid decrease in their weight-driven VFCs, and it gradually plateaus for later problems, which explains the plateauing of their weight-driven VFCs. Also, the net suppression is weaker along the orthogonal components than along the parallel components, which explains why the learning efficiency bottleneck develops along the orthogonal components. In Supplementary Note 1.1.3, we explored the dynamics of this cumulative suppression mechanism and determined that it resembles a stochastic process, with some problems suppressing a future problem's weight-driven VFC and others enhancing it (Supplementary Fig. 10c). However, the process exhibits a bias toward suppression, which produces the net suppressive effect. Modulation of this bias governs the learning-to-learn dynamics and time scale (Supplementary Fig. 10d).

Our results identify a novel neural mechanism of accumulating learning experience to progressively improve learning efficiency, despite the absence of a meta-learning mechanism. It relies on the accumulation of connection weight changes over learned problems to suppress the weight-driven VFC required to learn subsequent problems and, thus, accelerate their learning. The model progressively accelerates learning via (1) a gradual improvement in the efficiency with which weight changes contribute to the suppression of the weight-driven VFC in future problems (Supplementary Fig. 10a and Supplementary Note 1.1.3) and (2) a modulation of how consistently suppressive these contributions are (Supplementary Fig. 10d). Moreover, the fact that the weight-driven VFC primarily prevents uncharacteristic representational changes from developing when learning novel mappings (Fig. 5) helps elucidate the objective of this learning-to-learn mechanism: the accumulation of weight changes over early problems improves the invariance of the existing representations to having novel sample stimuli mapped onto them. This refines the model's ability to learn via representational reuse and elicits learning-to-learn.

Discussion

New information is easier to learn when contextualized by prior knowledge. This is facilitated by the instantiation of schemas[3,4], which are hypothesized to correspond to neocortically encoded knowledge structures. Learning-to-learn is a constructive consequence of the reciprocal influence between learning and schema tuning, whereby schema instantiation facilitates learning, and the assimilation of learned information into the schema improves its ability to facilitate future learning. To elucidate the underlying neurobiological basis, we trained an RNN model on a series of sensorimotor mapping problems, without meta-learning. Our main findings are three-fold. First, the network model exhibits accelerated learning that is quantified by an exponential time course, with a characteristic time constant and a plateau. This model prediction is supported by an ongoing experiment where monkeys displayed an exponential learning-to-learn time course while solving a series of arbitrary sensorimotor mapping problems (Peysakhovich et al., unpublished). Second, schema formation corresponds to the formation of a low-dimensional subspace of neural population activity, thereby bridging a psychological concept with a neural circuit mechanism. Third, rather than weight changes per se, it is imperative to examine weight-driven changes of the vector field to understand the behavior of a recurrent neural network as a dynamical system. These new insights can guide the analysis of neurophysiological data from behaving animals during learning-to-learn.

Our work revealed that learning-to-learn is a process with three time scales (Fig. 8). The fastest time scale governs the evolution of population activity over a single trial. Subspace decomposition of this activity showed that it encodes three latent variables. First, a mean decision component that is analogous to the condition-independent component identified in prefrontal and motor cortical activity[27,36]—it

https://doi.org/10.1038/s41593-023-01293-9

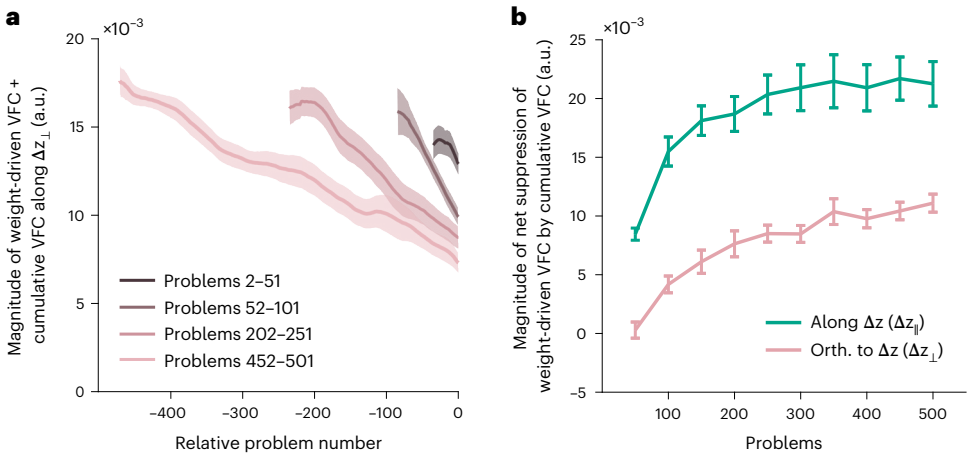

Fig. 7 | Accumulation of weight changes progressively improves invariance of existing representations to learning. a, Magnitude of VFC along the learned trajectory for a problem p due to the accumulation of (1) weight changes in problem p ($W^p - W^{p-1}$, relative problem = 0; weight-driven VFC) and (2) weight changes in each of the earlier problems, proceeding backwards to problem 2 ($W^p - W^{p-k-1}$ for $1 \le k \le p-2$, relative problem $-k$; cumulative VFC contributions). The curve for each problem measures the magnitude of change in the direction of its orthogonal weight-driven VFC component, smoothed with a 30-problem moving average filter. Plot summarizes the measurements for problems in four problem groups at different stages of learning-to-learn and demonstrates the suppressive effect of the cumulative VFC at each stage. **b**, Magnitude of net suppression for each problem p, due to the net weight changes between the start of problems 2 and p ($W^{p-1} - W^1$), summarized in 50-problem groups. The measure is presented separately for the VFC along the parallel (green) and orthogonal (pink) weight-driven VFC components. Magnitudes shown are the temporal mean of the unsigned (L^1-norm, **a**) and signed (**b**) projections onto the parallel/orthogonal weight-driven VFC components, averaged over both mappings in a problem. Plots reflect mean values over ten networks with different initial conditions, and shading/error bars indicate standard errors. a.u., arbitrary units; Orth., orthogonal.

encodes temporal aspects of the task in a trial-condition-invariant manner and explains most of the variance in population activity. Second, a residual decision component that encodes decisions and response choices. Third, a problem stimulus representation. The first two components collectively constitute low-dimensional decision representations that control fixation and response choices.

We found that these decision representations are shared across problems in an abstract form: the model reuses them to contextualize its neural and output responses to new sample stimuli and to generalize from previous solutions to newer ones. A manifold perturbation intervention showed that this reuse causes a stark improvement in learning efficiency. Therefore, the network not only abstracts commonalities across problems but also exploits them to facilitate learning[4,13,37]. This shows that the abstract decision representations constitute the neural basis of a sensorimotor mapping schema[4,7]. The abstraction of task variable-encoding and task structure-encoding neural representations and their reuse in consecutively learned association problems has indeed been observed in the prefrontal cortex and hippocampus[11,12,23].

The intermediate time scale governs the process of learning and spans the trials between the beginning and end of learning a single problem (Fig. 8). We studied learning with a novel measure of how connection weight changes (which model the effects of long-term synaptic plasticity (LTP)) influence population activity in an RNN—the weight-driven VFC. We found that this measure is more informative at assessing the effects of the connection weight changes than direct measurements of the weight changes: (1) it dissociates the contributions of the changes in different sets of connection weights more accurately than directly comparing their magnitudes; (2) its assessments are more interpretable, as they directly relate to the population activity; and (3) it isolates the contributions of the initial weights and the weight changes to the learning-induced changes in population activity. For these reasons, these techniques contribute to a growing set of methods aimed at overcoming the challenges of interpretability and explainability in RNNs[38,39], which hinder their adoption in neuroscience. In our analysis, these techniques were instrumental in identifying (1) why reusing existing representations improves learning

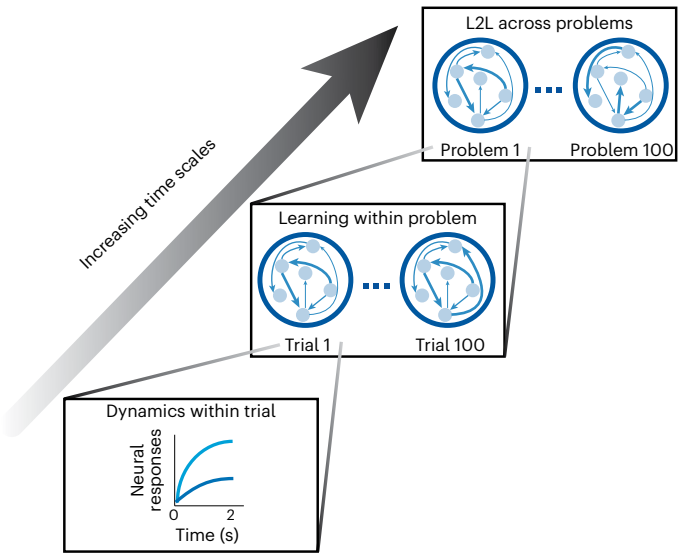

Fig. 8 | Learning-to-learn is a process with three time scales. The fastest time scale (bottom) governs the neural dynamics within a trial that drive output responses. The intermediate time scale (middle) governs the learning dynamics across trials within a problem; it ultimately produces the requisite weight-driven VFC, which results in the problem being learned. The slowest time scale (top) governs the dynamics of learning-to-learn across problems; it ultimately improves the invariance of existing representations to learning new problems, which results in asymptotic learning efficiency. L2L, learning-to-learn.

efficiency, (2) the relative contributions of this reuse versus the connection weight changes to learning and (3) the mechanism underlying learning-to-learn.

In the training RNN framework, the network is initialized with random weights, as a blank slate. In contrast, developmental experience shapes how new information is encoded even in the brain of a task-naive animal. This confounds direct comparisons between the

use of a learning algorithm and a known biological plasticity rule. Nevertheless, our findings regarding the benefits of representational reuse do not directly depend on our model's learning algorithm and may well be conserved under biologically plausible learning rules. Moreover, because our analysis techniques are independent of the underlying learning rules, they offer an approach to study learning and the properties of schema formation and reuse in models with biologically plausible learning rules. Our model further assumes that, after schema formation, new problems continue to be learned via LTP. Indeed, rapid learning of novel schema-consistent paired associates is prefrontal NMDA receptor dependent in rodents[40], suggesting that Hebbian neocortical synaptic plasticity is likely involved in schema-facilitated learning. However, the role of other forms of plasticity, such as intrinsic[41] and behavioral time scale[42] plasticity, has not been experimentally precluded. Further computational and experimental studies are required to determine their relative roles in this process.

At the slowest time scale, several problems are learned in succession with progressively improving efficiency, until asymptotic efficiency is realized (Fig. 8). This is the time scale of learning-to-learn. We showed that, consistent with macaque monkeys' behavior[19], our model's trials to criterion performance is well characterized by a decaying exponential function, which asymptotes at roughly 20 trials per problem. Consequently, our model suggests that learning-to-learn can emerge in animal models in the absence of explicit meta-learning (Supplementary Discussion 1.2.1).

We identified a novel mechanism for learning-to-learn, which relies on the accumulation of weight changes over learned problems to progressively improve the invariance of the existing representations to subsequent learning. An increase in this invariance suppresses the weight-driven VFCs required to learn new problems, which accelerates their learning. Interestingly, these cumulative improvements are stochastic in nature—the exponential improvement in learning efficiency stems from a modulation of the bias in this stochastic suppression of the weight-driven VFCs in future problems. These results also differentiate between schema-facilitated rapid learning and structure learning, which theorizes that the progressive learning acceleration arises from a refinement in the neural control of behavioral parameters[13] (Supplementary Discussion 1.2.2).

Crucially, our results offer experimentally verifiable predictions. First, the sensorimotor mapping schema is encoded by low-dimensional neural representations, which are shared across problems, and explain most of the variance in population activity. They encode shared task variables, including the task's temporal structure and the available choices. Second, the reuse of these representations to learn new problems speeds up learning; preventing this reuse with recently developed BCI interventions[24] should produce pronounced learning deficits. Third, population activity may undergo marked changes between the beginning and end of problem learning. However, across problems, these changes are restricted to a low-dimensional subspace of the activity. Fourth, the number of trials to learn a problem decreases exponentially with the number of previously learned problems. Taken together, our results provide insights into the neural substrate of a sensorimotor mapping schema, the reason for which its reuse markedly improves learning efficiency, and the neural mechanisms of structure learning that gives rise to learning-to-learn. In doing so, they elucidate the neural mechanisms of learning-to-learn and present novel techniques to analyze learning-to-learn in RNNs.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41593-023-01293-9.

References

1. Piaget, J. *The Language and Thought of the Child* (Harcourt Brace, 1926).
2. Bartlett, F.C. *Remembering: A Study in Experimental and Social Psychology* (Cambridge University Press, 1932).
3. Rumelhart, D. E. Schemata: the building blocks of cognition. in *Theoretical Issues in Reading Comprehension* 33–58 (Erlbaum Associates, 1980).
4. Gilboa, A. & Marlatte, H. Neurobiology of schemas and schema-mediated memory. *Trends Cogn. Sci.* **21**, 618–631 (2017).
5. Chi, M. T., Glaser, R. & Rees, E. Expertise in problem solving. https://www.public.asu.edu/~mtchi/papers/ChiGlaserRees.pdf (1982).
6. Harlow, H. F. The formation of learning sets. *Psychological Review* **56**, 51–65 (1949).
7. Lewis, P. A. & Durrant, S. J. Overlapping memory replay during sleep builds cognitive schemata. *Trends Cogn. Sci.* **15**, 343–351 (2011).
8. Behrens, T. E. J. et al. What is a cognitive map? Organizing knowledge for flexible behavior. *Neuron* **100**, 490–509 (2018).
9. Preston, A. R. & Eichenbaum, H. Interplay of hippocampus and prefrontal cortex in memory. *Curr. Biol.* **23**, R764–R773. (2013).
10. Wang, S.-H. & Morris, R. G. Hippocampal–neocortical interactions in memory formation, consolidation, and reconsolidation. *Annu. Rev. Psychol.* **61**, 49–79 (2010).
11. McKenzie, S. et al. Hippocampal representation of related and opposing memories develop within distinct, hierarchically organized neural schemas. *Neuron* **83**, 202–215 (2014).
12. Bernardi, S. et al. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell* **183**, 954–967 (2020).
13. Braun, D. A., Mehring, C. & Wolpert, D. M. Structure learning in action. *Behav. Brain Res.* **206**, 157–165 (2010).
14. Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning* 1126–1135 (PMLR, 2017).
15. Wang, J. X. et al. Prefrontal cortex as a meta-reinforcement learning system. *Nat. Neurosci.* **21**, 860–868 (2018).
16. Passingham, R. *The Frontal Lobes and Voluntary Action* (Oxford University Press, 1995).
17. Asaad, W. F., Rainer, G. & Miller, E. K. Neural activity in the primate prefrontal cortex during associative learning. *Neuron* **21**, 1399–1407 (1998).
18. Fusi, S., Asaad, W. F., Miller, E. K. & Wang, X.-J. A neural circuit model of flexible sensorimotor mapping: learning and forgetting on multiple timescales. *Neuron* **54**, 319–333 (2007).
19. Cromer, J. A., Machon, M. & Miller, E. K. Rapid association learning in the primate prefrontal cortex in the absence of behavioral reversals. *J. Cogn. Neurosci.* **23**, 1823–1828 (2011).
20. Bussey, T. J., Wise, S. P. & Murray, E. A. Interaction of ventral and orbital prefrontal cortex with inferotemporal cortex in conditional visuomotor learning. *Behav. Neurosci.* **116**, 703–715 (2002).
21. Petrides, M. Deficits on conditional associative-learning tasks after frontal- and temporal-lobe lesions in man. *Neuropsychologia* **23**, 601–614 (1985).
22. Stringer, C. et al. Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **364**, 255 (2019).
23. Zhou, J. et al. Evolving schema representations in orbitofrontal ensembles during learning. *Nature* **590**, 606–611 (2021).
24. Sadtler, P. T. et al. Neural constraints on learning. *Nature* **512**, 423–426 (2014).
25. Bao, P., She, L., McGill, M. & Tsao, D. Y. A map of object space in primate inferotemporal cortex. *Nature* **583**, 103–108 (2020).
26. Eacott, M. & Gaffan, D. Inferotemporal–frontal disconnection: the uncinate fascicle and visual associative learning in monkeys. *Eur. J. Neurosci.* **4**, 1320–1332 (1992).

27. Kobak, D. et al. Demixed principal component analysis of neural population data. *eLife* **5**, e10989 (2016).
28. Anderson, R. C., Spiro, R. J. & Anderson, M. C. Schemata as scaffolding for the representation of information in connected discourse. *Am. Educ. Res. J.* **15**, 433–440 (1978).
29. Rumelhart, D. E. & Norman, D. A. Accretion, tuning and restructuring: three modes of learning. https://www.dsoergel.com/UBLIS571DS-06.1a-1Reading10RumelhartAccretionTuningAndRestructuring.pdf (1978).
30. Thorndyke, P. W. & Hayes-Roth, B. The use of schemata in the acquisition and transfer of knowledge. *Cogn. Psychol.* **11**, 82–106 (1979).
31. Kaufman, M. T., Churchland, M. M., Ryu, S. I. & Shenoy, K. V. Cortical activity in the null space: permitting preparation without movement. *Nat. Neurosci.* **17**, 440–448 (2014).
32. Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering* 2nd edn (Taylor & Francis, 2016).
33. Vyas, S., Golub, M. D., Sussillo, D. & Shenoy, K. V. Computation through neural population dynamics. *Annu. Rev. Neurosci.* **43**, 249–275 (2020).
34. Long, P. M. & Sedghi, H. Generalization bounds for deep convolutional neural networks. In *International Conference on Learning Representations* (ICLR, 2020).
35. Gouk, H., Hospedales, T. M. & Pontil, M. Distance-based regularisation of deep networks for fine-tuning. In *International Conference on Learning Representations* (ICLR, 2021).
36. Kaufman, M. T. et al. The largest response component in the motor cortex reflects movement timing but not movement type. *eNeuro* **3**, ENEURO.0085-16.2016 (2016).
37. Tenenbaum, J. B., Kemp, C., Griffiths, T. L. & Goodman, N. D. How to grow a mind: statistics, structure, and abstraction. *Science* **331**, 1279–1285 (2011).
38. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–649 (2013).
39. Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F. & Ostojic, S. The role of population structure in computations through neural dynamics. *Nat. Neurosci.* **25**, 783–794 (2022).
40. Wang, S.-H., Tse, D. & Morris, R. G. Anterior cingulate cortex in schema assimilation and expression. *Learn. Mem.* **19**, 315–318 (2012).
41. Sehgal, M., Song, C., Ehlers, V. L. & Moyer, J. R. Jr. Learning to learn—intrinsic plasticity as a metaplasticity mechanism for memory formation. *Neurobiol. Learn. Mem.* **105**, 186–199 (2013).
42. Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S. & Magee, J. C. Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* **357**, 1033–1036 (2017).

Methods

RNN model

The RNN model comprises a fully connected population of N firing rate units with firing rates $\mathbf{r}$, receiving inputs from N_{in} input units with firing rates $\mathbf{u}$. Firing rates of the network units follow the dynamical equation:

$$\begin{aligned}\tau\dot{\mathbf{r}} &= -\mathbf{r} + f(W_{in}\mathbf{u} + W_{\text{rec}}\mathbf{r} + \mathbf{b_{rec}} + \boldsymbol{\zeta})\\ \tau_\zeta\dot{\boldsymbol{\zeta}} &= -\boldsymbol{\zeta} + \sqrt{2\tau_\zeta\sigma_{rec}^2}\boldsymbol{\xi}\end{aligned} \tag{1}$$

which expresses the leaky and non-linear integration of input ($W_{in}\mathbf{u}$) and recurrent ($W_{rec}\mathbf{r}$) currents. W_{in} (W_{rec}) is an $N \times N_{in}$ ($N \times N$) matrix of input (recurrent) connection weights, and $\tau = 100$ ms is the integration time constant that characterizes the slow decay of NMDA-receptor-mediated synaptic currents[43]. The f-I curve is modeled by a smooth rectification function:

$$f(x) = log(1 + e^x)$$

The bias term $\mathbf{b_{rec}}$ admits per-unit firing thresholds. Intrinsic background noise current is modeled by an Ornstein–Uhlenbeck process $\boldsymbol{\zeta}$ with time constant τ_ζ and variance σ_{rec}, where $\boldsymbol{\xi}$ represents the underlying independent white noise process with zero mean and unit variance.

Output responses are readout from the activity of the RNN units by N_{out} output units, $\mathbf{y}$, whose activity is given by

$$\mathbf{y} = g(W_{out}\mathbf{r} + \mathbf{b_{out}})$$

Here, W_{out} is an $N_{out} \times N$ output weight matrix; $\mathbf{b_{out}}$ is the bias of the output units; and $g(x_i) = exp(x_i) / \sum_{j=1}^{N_{out}} exp(x_j)$ is the softmax or normalized exponential function, which produces output unit activity that indicates the probability of generating each of the N_{out} response choices.

The model is simulated by temporal discretization of Eq. (1) with Euler's method as

$$\begin{aligned}\mathbf{r_t} &= (1-\alpha)\mathbf{r_{t-1}} + \alpha f(W_{in}\mathbf{u_t} + W_{rec}\mathbf{r_{t-1}} + \mathbf{b_{rec}} + \boldsymbol{\zeta_t})\\ \boldsymbol{\zeta_t} &= (1-\alpha_\zeta)\boldsymbol{\zeta_{t-1}} + \sqrt{2\alpha_\zeta\sigma_{rec}^2}\mathcal{N}(0, I)\end{aligned} \tag{2}$$

where the time-discretization step size is Δt, $\alpha = \Delta t/\tau$, $\alpha_\zeta = \Delta t/\tau_\zeta$, and $\mathcal{N}(0, I)$ is a random vector sampled from a Gaussian distribution with zero mean and identity covariance (I). In all figures, the network size $N = 100$, $\Delta t = 1$·ms, $\tau_\zeta = 2$·ms and $\sigma_{rec} = 0.05$. The magnitude of the network unit and input unit firing rates is measured as the L^2-norm of $\mathbf{r_t}$ and $\mathbf{u_t}$, respectively, and summarized by averaging over all time points in a trial.

Task structure

We trained the network model on a series of delayed sensorimotor association problems, one at a time. In each problem, the network learned a one-to-one correspondence between a pair of sample stimuli and a pair of motor responses. Each problem, therefore, comprised two trial types, one per stimulus–response pair. Each trial was 2 seconds in duration ($T = 2$) and started with a 500-ms sample epoch, followed by a 1-second delay epoch and ended with a 500-ms choice epoch. During the sample epoch, the network concurrently received inputs representing a fixation stimulus and one sample stimulus. During the delay epoch, it continued to receive only the fixation input. It received no inputs during the choice epoch. The model was required to maintain fixation during the sample and delay epochs and choose the appropriate motor response during the choice epoch. The model contained three output units ($N_{out} = 3$), two to report response choices and one for fixation. This trial structure, including the available response choices, remained fixed across problems.

Sample stimuli were represented by ten-dimensional unit length vectors (L^2-norm = 1). The two sample stimulus representations in a problem were drawn from a random Gaussian distribution with zero mean and identity covariance. They were then orthogonalized to avoid learning efficiency confounds stemming from the relative difficulty in learning to distinguish between more versus less correlated sample stimuli. The fixation input was a scalar with value $1/\sqrt{N_{in} - 1}$ when it was on and zero when off. Therefore, there was a total of $N_{in} = 11$ input units. Learning-to-learn was robustly observed even in the absence of the orthogonalization step; however, the variance in learning efficiency was higher. Qualitatively similar learning-to-learn performance was also observed with 200-dimensional sample stimulus representations and $N = 1{,}000$.

Each problem was learned over a sequence of trials, pseudorandomly sampled from the two trial types, until the average error on 50 consecutive trials fell below a criterion value (see the 'Network training' subsection). The learning efficiency for a problem was measured by the number of trials required to achieve this criterion. After a problem was learned, the model was transitioned to the next problem, wherein it learned to associate a new pair of pseudorandomly selected sample stimuli with the two motor responses.

Network training

A network was trained on a problem by updating its connection weights (W_{in}, W_{rec} and W_{out}), biases ($\mathbf{b_{rec}}$ and $\mathbf{b_{out}}$) and initial network state ($\mathbf{r_0}$), so that it could choose the desired response for each of the sample stimuli. These updates were generated by stochastic gradient descent—an optimization algorithm that incrementally updates a network's parameters at the end of each trial, based on the errors in the output unit responses during the trial. In contrast to standard RNN training practices, wherein model parameters are adjusted based on the average error from a batch of several trials and learning efficiency is measured by the number of trial batches to reach criterion performance, our training procedure closely matched established animal training protocols and allowed learning efficiency to be measured by the number of trials to criterion performance. The backpropagation through time (BPTT) algorithm was used to resolve temporal contingencies while computing parameter updates. We additionally applied the Adam optimizer[44] to enhance the efficacy of the updates. All networks were trained with a learning rate of 10^{-4}, except in Supplementary Fig. 1 where the learning rate was systematically varied. Adam decay rates for the first and second moment estimates were set to 0.3 and 0.999, respectively, and the moment estimates were reset at the beginning of each problem. The model implementation and parameter update computations were performed with TensorFlow[45] in the Python programming language and supported by the Numpy numerical computing library.

Before the first problem, a naive network's input weights in W_{in} were initialized with random values drawn from a Gaussian distribution with zero mean and variance $1/N_{in}$; the recurrent weights in W_{rec} were initialized with random values constrained by householder transformations such that the rows (and columns) of the initial recurrent weight matrix were orthogonal to each other and of unit length[46]. Initializing the recurrent weights in this manner allows gradients to be backpropagated more effectively. All other network parameters were initialized to zero. Upon transition to a new problem, all parameters retained their values. At initialization and throughout learning, the sign and sparsity of the weights and biases were not constrained. The initial network state was always restricted to non-negative values.

Network training was performed in a supervised setting, wherein the parameters were adjusted to minimize an objective function, $\mathcal{L}$, that included the errors in the model's output responses:

$$\mathcal{L}_{err} = \frac{1}{T - |D_{mask}|}\sum_{t \notin D_{mask}}\sum_{i=1}^{N_{out}} -\check{y}_{i,t}log(y_{i,t})$$

The error at each time step t was given by the cross-entropy of the probability distribution over responses generated by the network, $\mathbf{y_t}$, relative to pre-specified target responses, $\check{\mathbf{y}}_\mathbf{t}$. The total error for a trial, $\mathcal{L}_{err}$, was the mean of the per-time-step error taken over the trial duration T. This excluded a masking interval, D_{mask}, set to the first 100 ms of the choice epoch, which allowed for flexible reaction times. Networks were considered to have learned a problem when the average $\mathcal{L}_{err}$ over 50 consecutive trials of the problem fell below a criterion value of 0.005.

The objective of the training procedure was to minimize the sum of this error and auxiliary regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{err}} + \mathcal{L}_{reg,W_{in}} + \mathcal{L}_{reg,W_{out}} + \mathcal{L}_{reg,W_{rec}+\mathcal{L}_{reg,rate}}$$

The regularization terms included both weight and activity regularization to encourage solutions that generalized well[47,48] and generated stable network dynamics. We imposed L^2 regularization on the input and output weights as follows:

$$\mathcal{L}_{reg,W_{in}} = \frac{\beta_{W_{in}}}{N_{in}N}\sum_{i=1}^{N_{in}}\sum_{j=1}^{N}(W_{in}(j,i))^2$$

$$\mathcal{L}_{reg,W_{out}} = \frac{\beta_{W_{out}}}{N_{out}N}\sum_{i=1}^{N}\sum_{j=1}^{N_{out}}(W_{out}(j,i))^2$$

We observed that networks with a similar L^2 regularization of the recurrent weights were sensitive to the value of meta-parameter $\beta_{W_{rec}}$, particularly when the network size was large—small values of $\beta_{W_{rec}}$ produced unstable network dynamics during later problems, whereas large values hindered learning efficiency. The squared Frobenius norm of the recurrent weight matrix, which constitutes such an L^2 regularization, is given by:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(W_{rec}(j,i))^2 = \sum_{i=1}^{N}\sigma_i^2$$

where σ_i is the i-th singular value of the recurrent weight matrix W_{rec}.

An analysis of these singular values under conditions that led to unstable network dynamics revealed that their L^2-norm (that is, the square root of the righthand side of the equation above) remained roughly fixed over the course of learning several problems; however, their distribution changed considerably across problems—smaller singular values shrank, whereas larger singular values grew and ultimately resulted in unstable network responses to novel sample stimuli. We mitigated this by introducing an alternate form of recurrent weight regularization that penalized the magnitude of the first k singular values of W_{rec}:

$$\mathcal{L}_{reg,W_{rec}} = \frac{\beta_{W_{rec}}}{Nk}\sum_{i=1}^{k}\sigma_i^2$$

Finally, we imposed a homeostatic firing rate regularization:

$$\mathcal{L}_{reg,rate} = \beta_r\left|\frac{1}{NT}\sum_t\sum_{i=1}^{N}r_{i,t}^2 - h\right|$$

The meta-parameter h was set to zero for the first problem, effectively imposing an L^2 regularization of the recurrent unit firing rates as the first problem was learned. To avoid unrestrained growth or reduction in the firing rates while learning subsequent problems, the homeostatic setpoint h was then set to the mean squared firing rates averaged over the last 50 trials of the first problem. All networks were trained with $\beta_{W_{in}} = 10^{-4}$, $\beta_{W_{rec}} = 0.1$, $\beta_{W_{out}} = 0.1$, $k = 10$ and $\beta_r = 5\times10^{-4}$, except in Supplementary Fig. 1, where these hyperparameters were systematically varied.

Learning-to-learn performance characterization

A network's learning-to-learn performance (Fig. 1) was characterized by fitting a decaying exponential function to its number of trials to criterion $l(p)$ on problem p, as a function of the number of learned problems $p-1$:

$$l(p) = s_l\, exp\left(\frac{-(p-1)}{\tau_l}\right) + a_l$$

Here, a_l represents asymptotic learning efficiency; τ_l represents the time constant to achieve this asymptote; and s_l represents the improvement in learning efficiency between early and late problems. A large asymptote signifies poor learning-to-learn, whereas a large time constant signifies slow learning-to-learn. The three parameters of the function were fit with the Levenberg–Marquardt algorithm implemented by the `fit` function of MATLAB's Curve Fitting Toolbox. As a validation, these fits were compared to a moving average of the number of trials to criterion, calculated by MATLAB's `movmean` function, with a window size of 30 problems. The learning efficiency on the first problem was excluded from this analysis.

Subspace decomposition

We performed semi-supervised dimensionality reduction on the population activity, to determine how strongly and consistently the shared task structure is represented across problems (Fig. 2). First, we compiled a tensor $R_{k,t,j,i}$ of activity patterns generated by the population of firing rate units ($k \in [1,N]$) over time ($t \in [0,T]$), for the two response types ($j \in \{response_1, response_2\}$) across a group of 50 consecutively learned problems, ($i \in [p+1, p+50]$). Next, a semi-supervised dimensionality reduction extracted decision representations that are shared by the group as follows. Stimulus-specific and problem-specific representations for each response type are averaged out, or marginalized, across problems in the group:

$$R_{k,t,j,.} = < R_{k,t,j,i} >_i$$

Principal component analysis was performed on a concatenation of the resulting two trajectories in $R_{k,t,j,.}$. The loading vectors for the first m principal components were collected into an $N \times m$ loading matrix L_D. These vectors defined a basis for the decision subspace. To ensure that the decision subspace fully captured shared decision representations, the marginalized trajectories were not de-meaned before performing principal component analysis. Here, we set m to 4, as the first four principal components collectively explained at least 98% of the variance in the marginalized trajectories, in all the networks that we analyzed.

Next, an $N \times N$ projection matrix $P(Q)$ that projects population activity into the decision subspace (stimulus subspace) was defined as:

$$P = L_D L_D^T$$

$$Q = I - P$$

where I is the identity matrix. The decision components of the learned trajectories for problem $p+x$ ($x \in [1,50]$) were identified as:

$$R^d_{k1,t,j,i=p+x} = \sum_{k2=1}^{N} P(k1,k2)\, R_{k2,t,j,i=p+x}$$

and their stimulus components as:

$$R^s_{k1,t,j,i=p+x} = \sum_{k2=1}^{N} Q(k1,k2)\, R_{k2,t,j,i=p+x}$$

where $P(k1,k2)$ and $Q(k1,k2)$ represent the element in the $k1$-th row and $k2$-th column of the respective projection matrices. The decision

components were further decomposed into mean $\left(R^{dm}_{k,t,j,i=p+x}\right)$ and residual $\left(R^{dr}_{k,t,j,i=p+x}\right)$ decision components as:

$$R^{dm}_{k,t,.,i=p+x} = < R^{d}_{k,t,j,i=p+x} >_j$$

$$R^{dr}_{k,t,j,i=p+x} = R^{d}_{k,t,j,i=p+x} - R^{dm}_{k,t,.,i=p+x}$$

The net current from these components $R^{v}_{k,t,j,i=p+x} (v = \in \{s, dm, dr\})$ to an output unit o was computed as $\sum_{k=1}^{N} W^{p+x}_{out}(o,k) R^{v}_{k,t,j,i=p+x}$ where W^{p+x}_{out} is the output weight matrix learned in problem $p + x$. The dimensionality of any set of vectors (for example, population activity in the stimulus subspace) was approximated by its participation ratio[49], computed as $\frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2}$, where λ_i is the i-th eigenvalue of the covariance matrix of the vectors.

Manifold perturbations

To assess whether reuse of the decision representations improves learning efficiency, networks were trained on their second problem while constraining them in a manner that required the formation of new decision representations. The learning efficiency of such networks was compared to controls that were allowed to reuse existing decision representations while learning their second problem (Fig. 3).

A naive network was first trained on 50 problems, and the corresponding populations trajectories were used to identify its decision and stimulus subspaces. All network parameters were reset to their values at the end of the first problem. Then, its output weights were perturbed, and the network was trained on a new problem—that is, a second problem with respect to its parameters while barring the training procedure from changing its output weights. This procedure was repeated 50 times for each network, resetting its parameters, applying an independently chosen random perturbation to its output weights, freezing the output weights and training the network on a new sample stimulus pair each time. The output weights were subjected to one of three forms of perturbation. In the frozen readout condition, the output weights were unperturbed after the parameter reset. In D→S manifold perturbations, after the parameter reset, the output weights were perturbed to replace the overlap between the network's readout and decision subspaces with a corresponding overlap between its readout and stimulus subspaces:

$$W_{out,D\to S} = W_{out} - \sum_{i=1}^{4} W_{out} \mathbf{l}^{\mathbf{D}}_{\mathbf{i}} \mathbf{l}^{\mathbf{D}^T}_{\mathbf{i}} + \sum_{i=1}^{4} W_{out} \mathbf{l}^{\mathbf{D}}_{\mathbf{i}} \mathbf{l}^{\mathbf{S}^T}_{\sigma(\mathbf{i})}$$

where $W_{out,D\to S}$ is the perturbed output weight matrix; $l^D_i\left(l^S_i\right)$ is the i-th principal component loading vector of the decision (stimulus) subspace; and $\sigma()$ represents a random shuffle or permutation of the stimulus subspace principal component loading vectors. In S→S manifold perturbations, after the parameter reset, the output weights were perturbed to permute the overlap between the readout and stimulus subspaces:

$$W_{out,S\to S} = W_{out} - \sum_{i=1}^{4} W_{out} \mathbf{l}^{\mathbf{S}}_{\mathbf{i}} \mathbf{l}^{\mathbf{S}^T}_{\mathbf{i}} + \sum_{i=1}^{4} W_{out} \mathbf{l}^{\mathbf{S}}_{\mathbf{i}} \mathbf{l}^{\mathbf{S}^T}_{\sigma(\mathbf{i})}$$

Weight-driven and state-driven VFCs

Over the course of learning problem p, the model's parameters change from their values at the beginning of the problem—that is, their pre-learning values (W^{p-1}_{in}, W^{p-1}_{rec}, $\mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}$, W^{p-1}_{out}, $\mathbf{b}^{\mathbf{p-1}}_{\mathbf{out}}$ and $\mathbf{r}^{\mathbf{p-1}}_{0}$) to their values at the end of the problem—that is, their learned values (W^{p}_{in}, W^{p}_{rec}, $\mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}$, W^{p}_{out}, $\mathbf{b}^{\mathbf{p}}_{\mathbf{out}}$ and $\mathbf{r}^{\mathbf{p}}_{0}$). The difference between the learned and pre-learning values of the parameters quantify their change due to learning problem p (ΔW^{p}_{in}, ΔW^{p}_{rec}, $\Delta \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}$, ΔW^{p}_{out}, $\Delta \mathbf{b}^{\mathbf{p}}_{\mathbf{out}}$ and $\Delta \mathbf{r}^{\mathbf{p}}_{0}$) and are collectively referred to as ΔW^{p}.

Here, we present results relating the changes in these parameters to changes in the population's activity and dynamics. Although the results are presented in the context of temporally discretized dynamics, they may be readily extended to continuous time dynamics. Due to the parameter changes, the population activity in response to inputs $\mathbf{u}^{\mathbf{p}}_{\mathbf{t}}$ is altered from its pre-learning levels, $\mathbf{r}'^{\mathbf{p}}_{\mathbf{t}\in[\mathbf{0},\mathbf{T}]}$, to its learned ones, $\mathbf{r}^{\mathbf{p}}_{\mathbf{t}\in[\mathbf{0},\mathbf{T}]}$ (Fig. 4c, left). We derive an expression for this change in population activity, $\mathbf{z}^{\mathbf{p}}_{\mathbf{t}\in[\mathbf{0},\mathbf{T}]}$, in terms of the parameter changes. Based on the time-discretized model Eq. (2), we have:

$$\begin{aligned}
\mathbf{z}^{\mathbf{p}}_{\mathbf{t}} &= \mathbf{r}^{\mathbf{p}}_{\mathbf{t}} - \mathbf{r}'^{\mathbf{p}}_{\mathbf{t}} \\
&= \left[(1-\alpha)\,\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \alpha f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right)\right] - \\
&\quad \left[(1-\alpha)\,\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + \alpha f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] \\
&= \left[\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} - \mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}}\right] + \alpha\left[-\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right)\right] - \\
&\quad \alpha\left[-\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] \\
&= \left[\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} - \mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}}\right] + \alpha\left[-\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right)\right] - \\
&\quad \alpha\left[-\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] + \\
&\quad \alpha\left[-\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] - \\
&\quad \alpha\left[-\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right]
\end{aligned}$$

Rearranging the terms, we have:

$$\begin{aligned}
\mathbf{z}^{\mathbf{p}}_{\mathbf{t}} = \mathbf{z}^{\mathbf{p}}_{\mathbf{t-1}} + & \\
& \alpha\left[\left\{-\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right\} - \right. \\
& \quad \left.\left\{-\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right\}\right] + \\
& \alpha\left[f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right) - f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right]
\end{aligned} \tag{3}$$

This expression shows that the change in population activity emerges from an accumulation of activity change increments, $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$ (Fig. 4c, center):

$$\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}} = \mathbf{z}^{\mathbf{p}}_{\mathbf{t}} - \mathbf{z}^{\mathbf{p}}_{\mathbf{t-1}} \tag{4}$$

These increments are composed of two terms:

$$\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}} = \Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{s,t}} + \Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}} \tag{5}$$

The first term, $\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{s,t}}$, expresses the difference in the pre-learning vector field at the positions in state space along the learned $\left(\mathbf{r}^{\mathbf{p}}_{t-1}\right)$ and pre-learning $\left(\mathbf{r}'^{\mathbf{p}}_{\mathbf{t-1}}\right)$ trajectories (Fig. 4d, left). It is referred to as the state-driven VFC:

$$\begin{aligned}
\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{s,t}} = \alpha\left[\left\{-\mathbf{r}^{\mathbf{p}}_{t-1} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{t-1} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right\} - \right. \\
\left.\left\{-\mathbf{r}'^{\mathbf{p}}_{t-1} + f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}'^{\mathbf{p}}_{t-1} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right\}\right]
\end{aligned} \tag{6}$$

The second term, $\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}}$, expresses the change in the vector field at population states along the learned trajectory due to the parameter changes (Fig. 4d, center, and Fig. 4b, right). It is referred to as the weight-driven VFC:

$$\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}} = \alpha\left[f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right) - f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] \tag{7}$$

The weight-driven VFC stems from the change in the net afferent currents to the population, $\Delta\mathbf{Current}^{\mathbf{p}}_{\mathbf{w,t}}$ due to the parameter changes (Fig. 4b, left):

$$\begin{aligned}\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}} &= \alpha\left[f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right) - f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] \\ &= \alpha\left[f\left(\left(W^{p-1}_{in} + \Delta W^{p}_{in}\right)\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + \left(W^{p-1}_{rec} + \Delta W^{p}_{rec}\right)\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}}\right.\right. \\ &\quad \left.\left. + \left(\mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}} + \Delta\mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right)\right) - f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right] \\ &= \alpha\left[f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}} + \Delta\mathbf{Current}^{\mathbf{p}}_{\mathbf{w,t}}\right) - \right. \\ &\quad \left. f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right)\right]\end{aligned} \tag{8}$$

where $\Delta\mathbf{Current}^{\mathbf{p}}_{\mathbf{w,t}}$ is determined by ΔW^{p}_{in}, ΔW^{p}_{rec} and $\Delta\mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}$ as:

$$\Delta\mathbf{Current}^{\mathbf{p}}_{\mathbf{w,t}} = \Delta W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + \Delta W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \Delta\mathbf{b}^{\mathbf{p}}_{\mathbf{rec}} \tag{9}$$

The change in initial population state is defined as $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{0}} = \Delta\mathbf{r}^{\mathbf{p}}_{\mathbf{0}} = \mathbf{r}^{\mathbf{p}}_{\mathbf{0}} - \mathbf{r}^{\mathbf{p-1}}_{0}$. We omit the contribution of this change from our analyses, as it consistently showed a negligible effect on the evolution of the learned trajectory and the activity changes, across all problems and networks tested.

The contribution of the two VFC terms to the activity change increment, $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$, was measured by their magnitude along, or in the direction of, $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$ (Fig. 5a). This was computed by vector projection as:

$$\left|\Delta Field^{p}_{\mu,t}\right|_{\Delta z^{p}_{||}} = \Delta\mathbf{Field}^{\mathbf{p}}_{\boldsymbol{\mu},\mathbf{t}} \cdot \widehat{\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}}$$

where $\mu \in \{w, s\}$, $\cdot$ represents the dot product operator, and $\widehat{\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}}$ is the unit vector in the direction of $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$ $\left(\widehat{\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}} = \frac{\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}}{\left\|\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}\right\|_2}\right)$. Therefore, the VFC along $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$ is given by:

$$\Delta\mathbf{Field}^{p}_{\boldsymbol{\mu},\mathbf{t},\Delta\mathbf{z}^{\mathbf{p}}_{||}} = \left|\Delta Field^{p}_{\mu,t}\right|_{\Delta z^{p}_{||}} \widehat{\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}} \tag{10}$$

The remainder of each VFC term represents its components orthogonal to $\Delta\mathbf{z}^{\mathbf{p}}_{\mathbf{t}}$ (Fig. 5a):

$$\Delta\mathbf{Field}^{\mathbf{p}}_{\boldsymbol{\mu},\mathbf{t},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}} = \Delta\mathbf{Field}^{\mathbf{p}}_{\boldsymbol{\mu},\mathbf{t}} - \Delta\mathbf{Field}^{\mathbf{p}}_{\boldsymbol{\mu},\mathbf{t},\Delta\mathbf{z}^{\mathbf{p}}_{||}} \tag{11}$$

To compare the relative direction of the orthogonal components of the weight-driven and state-driven VFCs (Fig. 5a), we arbitrarily (but without loss of generality) chose the direction of $\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{s,t},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}}$ as the reference—signed magnitudes were computed by vector projection of $\Delta\mathbf{Field}^{\mathbf{p}}_{\boldsymbol{\mu},\mathbf{t},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}}$ onto a unit vector in the direction of $\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{s,t},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}}$.

The magnitude of change in the input and recurrent connection weights was measured by their Frobenius norm, $\left\|W^{p} - W^{p-1}\right\|_F = \sqrt{\sum_{i,j}\left(W^{p}(i,j) - W^{p-1}(i,j)\right)^2}$. Supplementary Methods 1.3.1 describes how we evaluate the contribution of changes in individual parameters (for example, input versus recurrent connection weights or recurrent weights from the decision versus stimulus subspace) to the change in the weight-driven VFC and the reciprocal interactions between decision and stimulus representations in sustaining the learned population trajectories.

Effects of weight change accumulation across problems

We measured the contribution of the weight changes elicited while learning problem $p-k$ (ΔW^{p-k}, for $1 \le k \le p-2$) to the cumulative VFC along the learned trajectory for problem p $\left(\Delta\mathbf{Field}^{\mathbf{p-k,p}}_{\mathbf{w,t}}\right)$ as:

$$\begin{aligned}\Delta\mathbf{Field}^{\mathbf{p-k,p}}_{\mathbf{w,t}} &= \alpha\left[f\left(W^{p-k}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-k}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-k}}_{\mathbf{rec}}\right) - \right. \\ &\quad \left. f\left(W^{p-k-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-k-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-k-1}}_{\mathbf{rec}}\right)\right]\end{aligned} \tag{12}$$

Then, the cumulative VFC due to the accumulation of weight changes across all the learned problems from $p-k$ to $p-1$ was given by:

$$\begin{aligned}\sum_{\mathbf{j=1}}^{\mathbf{k}}\Delta\mathbf{Field}^{\mathbf{p-j,p}}_{\mathbf{w,t}} &= \alpha\left[f\left(W^{p-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-1}}_{\mathbf{rec}}\right) - \right. \\ &\quad \left. f\left(W^{p-k-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p-k-1}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p-k-1}}_{\mathbf{rec}}\right)\right]\end{aligned} \tag{13}$$

The magnitude of cumulative VFC along the parallel ($\Delta\mathbf{z}^{\mathbf{p}}_{||}$) and orthogonal ($\Delta\mathbf{z}^{\mathbf{p}}_{\perp}$) components of the VFC for problem p were computed via vector projection of the cumulative VFC onto unit vectors in the direction of the VFC components. Specifically, given that the vectors $\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t},\Delta\mathbf{z}^{\mathbf{p}}_{||}}$ ($\Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}}$) are nearly one-dimensional across trial time t within problem p (Fig. 5c), we applied principal component analysis to find a single basis (unit-norm) vector, $\widehat{\Delta\mathbf{Field}}^{\mathbf{p}}_{\mathbf{w,e},\Delta\mathbf{z}^{\mathbf{p}}_{||}}$ $\left(\widehat{\Delta\mathbf{Field}}^{\mathbf{p}}_{\mathbf{w,e},\Delta\mathbf{z}^{\mathbf{p}}_{\perp}}\right)$, that accurately represents their shared direction during each non-overlapping 250-ms epoch, e, of the trial. The magnitude of the cumulative change along the parallel/orthogonal VFC component was given by:

$$\left|\sum_{\mathbf{j=1}}^{\mathbf{k}}\Delta Field^{p-j,p}_{w,t}\right|_{\Delta z^{p}_{\mu}} = \left|\left(\sum_{\mathbf{j=1}}^{\mathbf{k}}\Delta\mathbf{Field}^{\mathbf{p-j,p}}_{\mathbf{w,t}}\right) \cdot \widehat{\Delta\mathbf{Field}}^{\mathbf{p}}_{\mathbf{w,e},\Delta\mathbf{z}^{\mathbf{p}}_{\mu}}\right| \tag{14}$$

where $\mu \in \{||, \perp\}$, and time t lies within the interval of epoch e. The magnitudes of cumulative VFC contribution by individual problems along the parallel/orthogonal VFC component $\left(\left|\Delta Field^{p-k,p}_{w,t}\right|_{\Delta z^{p}_{\mu}}\right)$ were computed similarly.

The signed cumulative VFC and per-problem cumulative VFC contributions in Supplementary Fig. 10c were calculated as above but without taking the absolute value on the righthand side.

The per-trial magnitude of the cumulative VFC contribution by problem $p-k$ to problem p was calculated as $\frac{\left|\Delta Field^{p-k,p}_{w,t}\right|_{\Delta z^{p}_{\mu}}}{l(p-k)}$, where $l(\mathrm{p}-k)$ is the trials to criterion for problem $p-k$. The sum of the magnitudes of the cumulative VFC contributions to problem p was calculated as $\sum_{j=1}^{p-2}\left|\Delta Field^{p-j,p}_{w,t}\right|_{\Delta z^{p}_{\mu}}$.

The magnitude of net suppression of problem p's weight-driven VFC along its parallel/orthogonal component is defined as the net suppression in the direction of the corresponding component due to net weight changes between the start of problems 2 and p. It was computed from the total VFC along the learned trajectory for problem p since the start of problem 2. Let $\Delta\mathbf{Field}^{\mathbf{total,p}}_{\mathbf{w,t}}$ represent this total VFC at time t:

$$\Delta\mathbf{Field}^{\mathbf{total,p}}_{\mathbf{w,t}} = \sum_{\mathbf{j=1}}^{\mathbf{p-2}}\Delta\mathbf{Field}^{\mathbf{p-j,p}}_{\mathbf{w,t}} + \Delta\mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}}$$

Then, the total change along the parallel/orthogonal VFC component was given by:

$$\Delta F^{total,p}_{w,t,\Delta z^{p}_{\mu}} = \Delta\mathbf{Field}^{\mathbf{total,p}}_{\mathbf{w,t}} \cdot \widehat{\Delta\mathbf{Field}}^{\mathbf{p}}_{\mathbf{w,e},\Delta\mathbf{z}^{\mathbf{p}}_{\mu}}$$

We applied a sign correction to this quantity to ensure that its temporal mean is always positive. This allowed us to accurately calculate the net suppression. After sign correction, $\Delta F^{total,p}_{w,t,\Delta z^{p}_{\mu}}$ becomes:

$$\widetilde{\Delta F}^{total,p}_{w,t,\Delta z^{p}_{\mu}} = sgn\left(\Delta F^{total,p}_{w,.,\Delta z^{p}_{\mu}}\right)\Delta F^{total,p}_{w,t,\Delta z^{p}_{\mu}}$$

where $\Delta F^{total,p}_{w,.,\Delta z^{p}_{\mu}}$ represents the temporal mean of $\Delta F^{total,p}_{w,t,\Delta z^{p}_{\mu}}$ over time t within a trial, and $sgn()$ represents the signum function. Similarly, the weight-driven VFC for problem p along its parallel/orthogonal components was given by:

$$\Delta F^{p}_{w,t,\Delta z^{p}_{\mu}} = \Delta \mathbf{Field}^{\mathbf{p}}_{\mathbf{w,t}} \cdot \widehat{\Delta \mathbf{Field}}^{\mathbf{p}}_{\mathbf{w,e,\Delta z^{p}_{\mu}}}$$

Then, the magnitude of net suppression along the parallel/orthogonal VFC component for problem p was:

$$\widetilde{\Delta F}^{net,p}_{w,t,\Delta z^{p}_{\mu}} = \widetilde{\Delta F}^{total,p}_{w,t,\Delta z^{p}_{\mu}} - \Delta F^{p}_{w,t,\Delta z^{p}_{\mu}} \quad (15)$$

The progression of this quantity over the learning-to-learn time course can be described in terms of the number of previously learned problems. We note that the temporal mean of the magnitude of the weight-driven VFC along its parallel/orthogonal component $\left(\Delta F^{p}_{w,.,\Delta z^{p}_{\mu}}\right)$ decays exponentially from problem 2 onwards until an asymptotic value b_{μ} is converged upon (as in Fig. 6b). This decay may be expressed as:

$$\left(\Delta F^{p}_{w,.,\Delta z^{p}_{\mu}} - b_{\mu}\right) = \left(\Delta F^{2}_{w,.,\Delta z^{2}_{\mu}} - b_{\mu}\right) r_{\mu}^{p-2}$$

for an appropriate base $r_{\mu} < 1$. Taking the temporal mean of Eq. (15) over trial time t, we have:

$$\begin{aligned}
\widetilde{\Delta F}^{net,p}_{w,.,\Delta z^{p}_{\mu}} &= \widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - \Delta F^{p}_{w,.,\Delta z^{p}_{\mu}} \\
&= \widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - \left(\Delta F^{p}_{w,.,\Delta z^{p}_{\mu}} - b_{\mu} + b_{\mu}\right) \\
&= \widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - \left(\Delta F^{p}_{w,.,\Delta z^{p}_{\mu}} - b_{\mu}\right) - b_{\mu} \\
&= \widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - \left(\Delta F^{2}_{w,.,\Delta z^{2}_{\mu}} - b_{\mu}\right) r_{\mu}^{p-2} - b_{\mu}
\end{aligned}$$

Rearranging, we have:

$$\widetilde{\Delta F}^{net,p}_{w,.,\Delta z^{p}_{\mu}} = \left(\widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - b_{\mu}\right) - \left(\Delta F^{2}_{w,.,\Delta z^{2}_{\mu}} - b_{\mu}\right) r_{\mu}^{p-2} \quad (16)$$

This equation expresses the progression of the magnitude of net suppression over the learning-to-learn time course and determines its shape as a function of the number of previously learned problems (Fig. 7b). Note that, when the first term $\left(\widetilde{\Delta F}^{total,p}_{w,.,\Delta z^{p}_{\mu}} - b_{\mu}\right)$ is roughly constant across learning-to-learn stages (as we found by measurement), the magnitude of net suppression is given by an inverted exponential function.

Finally, we determined the relative contributions of the cumulative input versus recurrent weight changes to the cumulative VFC along the orthogonal VFC component (Supplementary Fig. 8c). To do so, we calculated the cumulative VFC for problem p solely due to the accumulation of input weight changes elicited by previously learned problems as:

$$\begin{aligned}
&\sum_{\mathbf{j=1}}^{\mathbf{k}} \Delta \mathbf{Field}^{\mathbf{p-j,p}}_{\mathbf{w_{in},t}} \\
&= \alpha\left[f\left(W^{p}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right) - f\left(W^{p-k-1}_{in}\mathbf{u}^{\mathbf{p}}_{\mathbf{t}} + W^{p}_{rec}\mathbf{r}^{\mathbf{p}}_{\mathbf{t-1}} + \mathbf{b}^{\mathbf{p}}_{\mathbf{rec}}\right)\right]
\end{aligned}$$

The cumulative VFC solely due to recurrent weight changes was calculated similarly. Both quantities were then projected onto the basis vector for the orthogonal VFC components in problem p (as in Eq. (14)), to compare their contributions along this component.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Data files, including pre-trained networks, are available for further analyses on GitHub (https://github.com/xjwanglab/learning-2-learn) in Python and MATLAB readable formats.

Code availability

All training and analysis codes are available on GitHub (https://github.com/xjwanglab/learning-2-learn).

References

43. Wang, X.-J. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron* **36**, 955–968 (2002).
44. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *International Conference on Learning Representations* (ICLR, 2015).
45. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. *USENIX Symposium on Operating Systems Design and Implementation* **16**, 265–283 (2016).
46. Stewart, G. W. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal.* **17**, 403–409 (1980).
47. Krogh, A. & Hertz, J. A. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems* 950–957 (NeurIPS, 1991).
48. Merity, S., McCann, B. & Socher, R. Revisiting activation regularization for language RNNs. In *International Conference on Machine Learning's Workshop on Learning to Generate Natural Language* (ICML, 2017).
49. Gao, P. et al. A theory of multineuronal dimensionality, dynamics and measurement. Preprint at https://www.biorxiv.org/content/10.1101/214262v2 (2017).

Acknowledgements

We thank A. L. Fairhall, I. Skelin, J. J. Lin, B. Doiron, G. R. Yang, N. Y. Masse, U. P. Obilinovic, L. Y. Tian, D. V. Buonomano, J. Jaramillo, J. E. Fitzgerald and H. Sompolinksy for fruitful discussions and Y. Liu, K. Berlemont, A. Battista and P. Theodoni for critical comments on the manuscript. This work was supported by National Institute of Health U-19 program grant 5U19NS107609-03 and Office of Naval Research grant N00014-23-1-2040.

Author contributions

B.P., D.J.F., E.A.B. and X.-J.W. designed the study. V.G. performed the research. V.G. and X.-J.W. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41593-023-01293-9.

Correspondence and requests for materials should be addressed to Xiao-Jing Wang.

Peer review information *Nature Neuroscience* thanks the anonymous reviewers for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Xiao-Jing Wang

Last updated by author(s): Jan 30, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Data was generated via simulation in Python 2.7 (with Numpy 1.16.5) wherein networks were trained with the Tensorflow 1.15 package that implements the stochastic gradient descent algorithm and ADAM optimizer. All training and analysis code is publicly available on Github (https://github.com/xjwanglab/learning-2-learn).
Data analysis	Data analysis was performed with Matlab 20. Code is publicly available on Github (https://github.com/xjwanglab/learning-2-learn).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Pre-trained networks are stored in a Google Drive folder with its link provided on the Github code repository (https://github.com/xjwanglab/learning-2-learn). Data files are provided in Python and Matlab readable formats for further analyses.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample sizes were determined by computational feasibility. However, they are on par with those reported in previous publications (Yang et al., Nature Neuroscience, 2019; Masse et al., Nature Neuroscience, 2019).
Data exclusions	None.
Replication	All analysis code was written from scratch at least twice. Model tested with multiple initial conditions. In all cases, we were able to reproduce findings.
Randomization	Samples for each group were determined based on parameter settings. For each parameter setting, multiple networks with different initial conditions were tested. The same set of seeds were used across parameter settings.
Blinding	Data collection and analysis were not performed blind to the conditions of the experiments. Since the analysis was performed on computer models rather than human or animal subjects, blinding was deemed unnecessary. Moreover, since causal analyses of neural mechanisms are challenging to perform with blinding.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging