nature neuroscience

Article

https://doi.org/10.1038/s41593-024-01852-8

Neural mechanisms of relational learning and fast knowledge reassembly in plastic neural networks

Received: 4 September 2023

Accepted: 15 November 2024

Published online: 15 January 2025

Check for updates

Thomas Miconi[1,2] & Kenneth Kay[3,4,5]

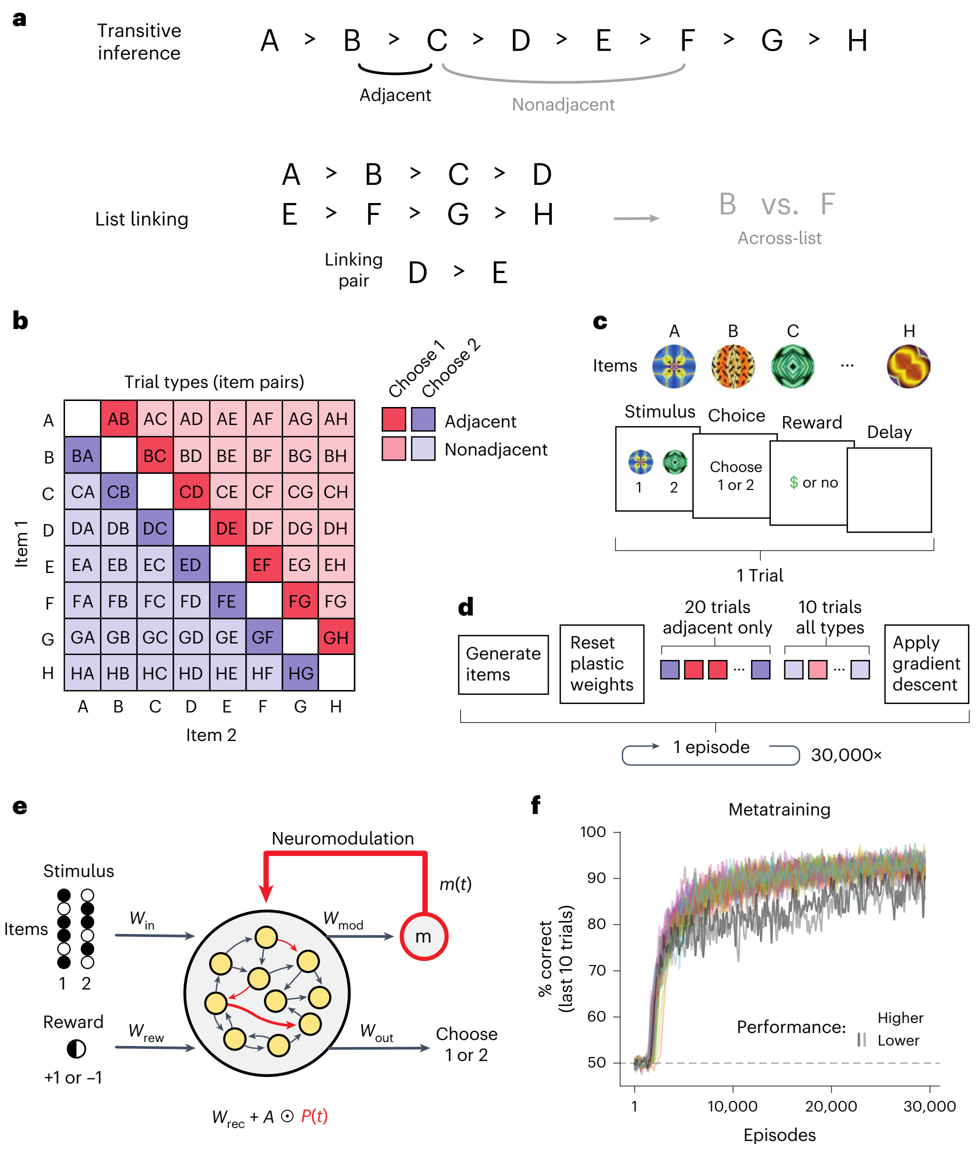
Humans and animals have a striking ability to learn relationships between items in experience (such as stimuli, objects and events), enabling structured generalization and rapid assimilation of new information. A fundamental type of such relational learning is order learning, which enables transitive inference (if A > B and B > C, then A > C) and list linking (A > B > C and D > E > F rapidly 'reassembled' into A > B > C > D > E > F upon learning C > D). Despite longstanding study, a neurobiologically plausible mechanism for transitive inference and rapid reassembly of order knowledge has remained elusive. Here we report that neural networks endowed with neuromodulated synaptic plasticity (allowing for self-directed learning) and identified through artificial metalearning (learning-to-learn) are able to perform both transitive inference and list linking and, further, express behavioral patterns widely observed in humans and animals. Crucially, only networks that adopt an 'active' solution, in which items from past trials are reinstated in neural activity in recoded form, are capable of list linking. These results identify fully neural mechanisms for relational learning, and highlight a method for discovering such mechanisms.

How do we gain broad knowledge from limited experience? Humans and animals can generalize and learn rapidly from limited experience, yet how these abilities are implemented in the brain remains an open question[1–3]. A possibly fundamental basis is the learning of relations between different experiences (such as 'stronger than', 'next to', 'same as' and 'part of') because relations specify a particular structure for subsequent generalization and inference. Additionally, such relational learning is thought to enable the construction and rapid modification of internal models of the world—variously termed 'relational memory', 'schemas' and 'cognitive maps'—which are increasingly recognized as essential to cognition[4,5].

One fundamental type of relational learning is order learning, which is broadly applicable to a range of concepts such as space, time, rank and number[6,7]. Critically, understanding order confers the ability to perform transitive inference, the ability to infer relative order between items not previously observed together. For example, after learning to choose between two stimuli that are 'adjacent' in an underlying ordered series—or 'list' (A > B, B > C, C > D, etc., where '>' designates 'chosen over')—humans and animals can choose correctly on 'nonadjacent' pairs not previously seen (A > C, B > D, etc.). Transitive inference has been observed in a wide range of species including humans, monkeys, rodents, birds and insects[6,8].

Remarkably, animals and humans can also rapidly learn a global ordering across separately learned lists, when presented with limited new information. That is, after separately learning lists (for example, A > B > C and D > E > F), participants that learn a linking pair (here C > D)

[1]ML Collective, San Francisco, CA, USA. [2]The Astera Institute, Berkeley, CA, USA. [3]Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University, New York City, NY, USA. [4]Center for Theoretical Neuroscience, Columbia University, New York City, NY, USA. [5]Grossman Center for the Statistics of Mind, Columbia University, New York City, NY, USA. e-mail: thomas.miconi@gmail.com; kaykenneth@gmail.com



	A	B	C	D	E	F	G	H
A		AB	AC	AD	AE	AF	AG	AH
B	BA		BC	BD	BE	BF	BG	BH
C	CA	CB		CD	CE	CF	CG	CH
D	DA	DB	DC		DE	DF	DG	DH
E	EA	EB	EC	ED		EF	EG	EH
F	FA	FB	FC	FD	FE		FG	FG
G	GA	GB	GC	GD	GE	GF		GH
H	HA	HB	HC	HD	HE	HF	HG	

Fig. 1 | Tasks, model and overall performance. a, Schematic of task paradigms. Letters (A, B, C, etc.) refer to arbitrary stimuli (items). In transitive inference, participants learn to choose between 'adjacent' pairs of items (such as B over C) in accordance with an underlying ordering (list) and are subsequently tested on nonadjacent pairs (such as C over F). In list linking, participants first learn to choose between adjacent pairs of items within two separate lists (such as A to D and E to H), then encounter a 'linking pair' (here D versus E), after which participants are tested on across-list pairs (such as B versus F). **b**, Table of trial types. Each trial type is defined by the pair of items presented in the trial (identity of item 1 and item 2). The correct choice is the higher-ranked item (item 1 versus item 2). **c**, Trial structure. Each trial is composed of four time steps as follows: stimulus presentation (stimulus), response (choice), feedback signal (reward) and delay. For illustration, each step is depicted as would be presented in a real-world experiment, with random fractal images as items; actual stimuli are binary vectors, randomly generated for each episode. **d**, Episode structure. Each episode consists of 20 trials with only adjacent pairs, followed by 10 trials with all possible pairs. Plastic weights are reset between episodes. **e**, Schematic of the neural model. The model is a recurrent network augmented with plastic weights controlled by self-generated neuromodulation. See 'Main' and Methods for full description. **f**, Model performance across metatraining. Plotted is the mean performance on the last ten trials of episodes (30,000 total) for 30 separate metatraining runs. Curves smoothed with a boxcar filter of width 10. Here two runs show consistently lower performance, suggestive of a different task solution.

can then immediately infer across-list pairs (for example, B > E). This ability, known as 'list linking'[8–10], demonstrates not only rapid assimilation of new information but also the fast 'reassembly'[10] of existing knowledge—after the presentation of the linking pair, participants must somehow reorganize their existing representations of previously learned list items not in the linking pair.

Numerous models of order learning have been proposed to explain transitive inference and list linking (see refs. 8,11 for reviews). However, currently, no biologically plausible neural model of order learning reproduces the various behavioral patterns commonly reported in experimental work[11]. Recent experimental findings suggest that relational learning reorganizes neural representations of linked items, potentially enabling inferential responses as a result[10,12], but leave open the question of how this reorganization occurs. In modeling work, neural networks can be explicitly trained (through an artificial optimization algorithm) to learn a given list or set of lists, including list linking; these networks exhibit internal representations and behavioral patterns consistent with experimental results[10,13–15]. However, because these approaches use hand-designed, nonbiological learning algorithms (usually based on backpropagation), or leave the learning process largely unspecified, they do not explain how order learning can be implemented in biologically plausible neural processes.

To address this challenge, here we take a different approach—rather than use a prespecified learning algorithm to train a neural system on one (or several) particular lists, we instead metatrain a learning neural system to be able to learn arbitrary new lists. Building upon previous work in metalearning, or 'learning-to-learn'[16–20], we metatrained neural networks endowed with biologically plausible (Hebbian) synaptic plasticity and self-controlled neuromodulation. These networks are able to actively modify their own connectivity in response to external input (such as sensory stimuli and rewards), thereby enabling autonomous, self-directed learning across trials. Importantly, the learning algorithm that these networks employ is not manually specified in advance, but instead entirely 'discovered' by the metatraining process.

Results

Model overview

We metatrain a recurrent neural network, endowed with synaptic plasticity and neuromodulation, to be able to autonomously learn an arbitrary implicit serial order for a set of arbitrary stimuli, over the course of several trials in which pairs of stimuli are presented, following the classic paradigm of transitive inference[6,8–11]. Schematics of the task and model are shown in Fig. 1a–e.

The task is organized into episodes, each of which is composed of a number of trials (Fig. 1b–d). In each episode, the agent is tasked with learning an implicit ordering over completely new random stimuli (items A, B, C, etc.; Fig. 1a). The stimuli are high-dimensional binary vectors (size = 15; we obtained similar results with one-hot vectors), randomly generated anew for each episode. The number of items in the list to be learned varies from episode to episode between four and nine (inclusive); all results below use eight items. Each episode consists of 30 trials, where each trial consists of the simultaneous presentation of two stimuli, a binary response by the agent (choose item 1 or item 2), and a binary feedback signal $R(t)$ indicating whether the response was correct or not (that is, whether the chosen stimulus was in fact the higher ranking of the two in the overall ordering).

The first 20 trials of an episode include only adjacent pairs, that is, pairs of stimuli with adjacent ranks in the series. The last ten trials include all possible pairs (excluding identical pairs such as AA or BB), unless specified otherwise. Performance in a given episode is assessed as the proportion of correct responses over the last ten trials.

Within each episode, the agent undergoes synaptic changes (plasticity), gated by a self-generated modulatory signal (Fig. 1e and Methods). From these synaptic changes, a successful agent would be able to learn the correct ordering of all stimuli over the course of the episode. To generate such agents, after each episode, we apply gradient descent to the structural parameters of the network (the base weights and plasticity parameters, as well as other parameters; see below), to improve within-episode plasticity-based learning. The loss optimized by gradient descent is the total reward obtained over the whole episode.

The gradients are computed by a simple reinforcement learning algorithm, namely, Advantage Actor Critic (A2C)[21], which is readily interpretable as modeling dopamine-based learning in the brain[19]. See Methods and Supplementary Note 5 for details.

Metalearning discovers solutions for transitive inference

Across multiple runs, the metatraining procedure described above consistently generated a high-performing learning agent (Fig. 1f). Subsequent analysis of trained networks indicated that all runs that reach the higher performance level yield a similar solution, which is presented below. Interestingly, some runs yielded agents that performed at relatively lower levels (Fig. 1f, gray traces), suggestive of a different solution in these agents. We return to these lower-performing agents in a subsequent section ('A suboptimal solution cannot perform list linking'); in the following sections, we analyzed a single network representative of the higher-performing agents (unless stated otherwise).

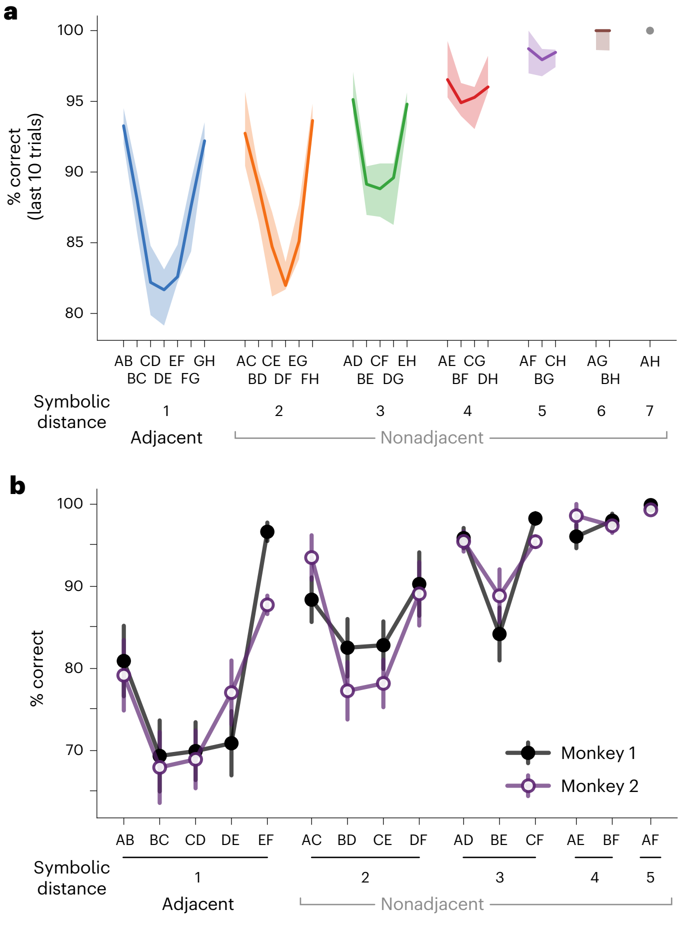


Fig. 2 | Reproduction of experimentally observed behavioral patterns. **a**, Performance (% correct) by trial type. Trial types (item pairs; Fig. 1b) are arranged according to the difference in rank between items ('symbolic distance'). Network responses were taken exclusively from the last ten trials of episodes, and for 2,000 separately run episodes split into ten subsets of 200 episodes each. Solid lines and shaded areas indicate median and interquartile range of mean performance for each pair across subsets. Note the higher performance for higher symbolic distance pairs (symbolic distance effect), and higher performance for pairs that include highest- or lowest-ranked items (here A or H (end-anchor effect)). **b**, Experimental data from monkeys (redrawn with permission from Brunamonti et al. (Fig. 1 of ref. 22)), displaying the same behavioral patterns (mean and s.e. of the mean for success rate over 30 sessions, each with at least 14 test trials for each pair, for either participant). See also Fig. 2 of ref. 11 and human data in Supplementary Fig. 5 of ref. 14.

Metatrained networks reproduce experimentally observed behavioral patterns

To assess the behavior of a successful learning network, we ran a single episode (20 learning trials with only adjacent pairs, followed by 10 test trials with all possible pairs) with eight stimuli (A to H) randomly generated for that episode. Figure 2a reports performance on the last ten (test) trials of this episode, separately for each pair, with pairs arranged according to 'symbolic distance', that is, the absolute difference in rank between items in the pair.

The network expressed two classic behavioral patterns characteristic of humans and animals in transitive inference experiments[8,11]. First, we observed the so-called symbolic distance effect[6,8,10,11,22]—performance is higher for the pairs with higher symbolic distance (upward trend from left to right, Fig. 2a; compare with Fig. 2b, showing monkey experimental data). In particular, performance is the lowest for adjacent pairs (AB, BC, etc.). This effect is especially notable because the first 20 trials involve only adjacent pairs (as done in experiments, serving as a 'training set' for the task)—performance

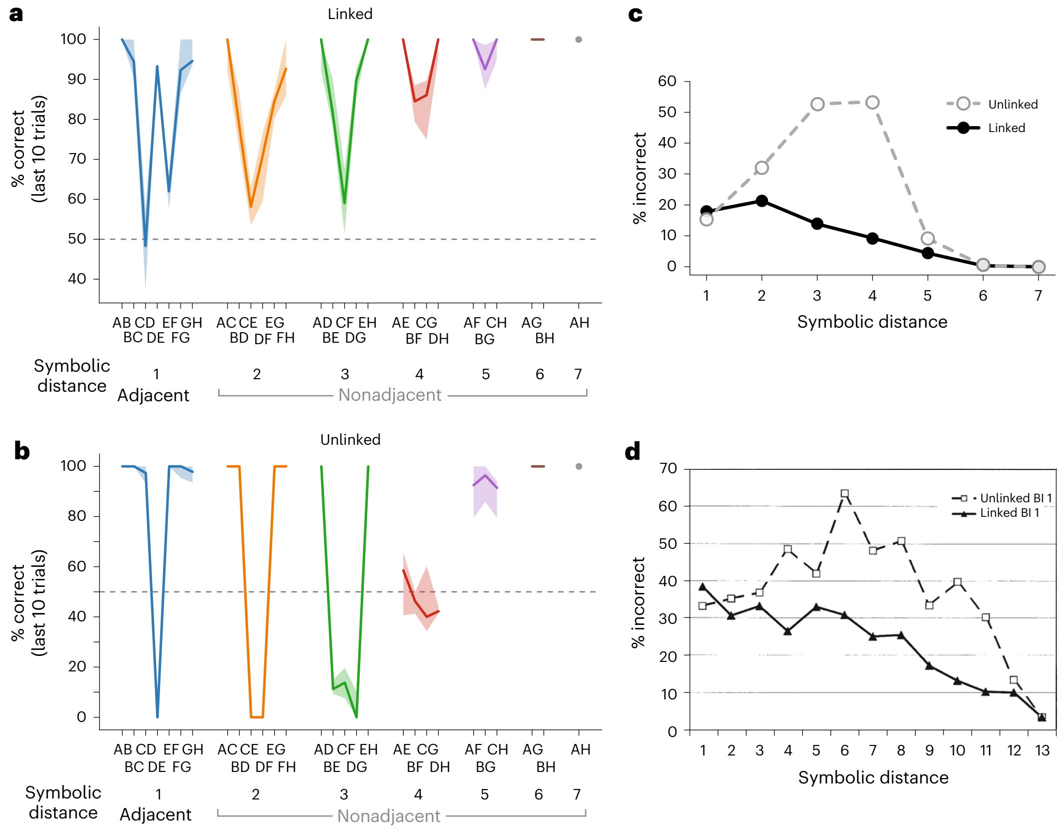


Fig. 3 | List-linking performance and comparison to experimental data. **a**, Performance (% correct) by trial type for episodes testing list linking. The network was presented with trials for learning ABCD, EFGH and DE, then tested on a single trial of any type (any item pair). Performance was measured solely for the final test trial. Conventions are the same as in Fig. 2. Note the relatively lower performance for pairs immediately adjacent to the linking pair (here CD and EF), as reported in experiments[9]. **b**, Performance in unlinked ('sham') condition. Instead of the linking pair DE, the network was presented with EF. **c**, Symbolic distance effect across lists, in linked and unlinked ('sham') conditions. Data are the same as in **a** and **b**. Note that the linked condition produces a symbolic distance effect that is largely monotonic, while the sham condition (unlinked) produces a bump in error for intermediate symbolic distances. **d**, Monkey experimental data, following the basic conventions in **c**, from ref. 24. See also Fig. 4b of ref. 10 for human experimental data.

is thus worse on the 'training set' compared to the 'test set', contrary to standard expectation.

Second, we observe an end-anchor effect[6,11,14,22] (also known as 'serial position effect')—performance is consistently higher for item pairs that involve the highest- or lowest-ranked items rather than other pairs of equivalent symbolic distance (for example, performance on AC and FH is higher than performance on CE or DF). This effect is seen as a U shape for sets of pairs of the same symbolic distance (Fig. 2a compared with Fig. 2b; refs. 8,11). Thus, the network successfully demonstrates transitive inference and reproduces behavioral patterns consistently observed in animal and human experiments.

Notably, in additional experiments, we found that the network is robust to so-called 'massed presentation' of one single pair, which is known to disrupt performance in certain learning models of transitive inference but not in living subjects[11,23] (see Supplementary Note 9 for details).

Metatrained networks rapidly reassemble existing knowledge

Monkeys and humans can rapidly 'link' separately learned lists after learning an item pair relating the two lists. That is, after learning A > B > C > D and E > F > G > H separately, and then learning D > E, they can quickly infer ordering across the entire joint list (C > F, B > G, etc.)[8–11]. This list linking ability implies that the presentation of a pair can affect the subjective ranking not just of items in the linking pair (here D and E) but also of other previously learned items not shown in the current trial.

We ran the metatrained network on ten trials using adjacent pairs from ABCD, then ten trials using adjacent pairs from EFGH, and then finally four trials with D and E. Then we estimated performance on a single 'test' trial, which could use any pair from the whole ABCDEFGH ordering. This was repeated over 2,000 runs, again with different randomly generated stimuli for each run. Results for the last 'test' trial are shown in Fig. 3a. Examination of performance on pairs including items from both sublists (for example, CE, CF, BG, etc.) confirmed that the network successfully linked the two lists into a coherent global ordering. Interestingly, performance was consistently poor for the pairs immediately adjacent to the linking pair, especially from the earlier-learned list (here CD), a pattern reported in monkey experiments (Table 2 of ref. 9).

A control experiment in which a different, nonlinking pair (EF) was shown for one trial only (instead of four trials with the linking pair DE, as above) produced no evidence of list linking (unlinked or 'sham' condition; Fig. 3b). In this case, performance was far below chance on pairs consisting of items whose rank within their respective lists conflicted with their overall rank in the global list (for example, for pair CE, E has high rank within its own list EFGH, while C has low rank within-list ABCD, yet C is higher than E in the overall combined list ABCDEFG). This suggests that the rank of an item generalizes across unlinked lists; such transfer of rank across lists is also observed in experiments[11,24]. This effect produces a characteristic bump in error rate for intermediate symbolic distances with nonlinked lists but not with linked lists (because pairs where within-list and across-list ranks conflict are more likely to have intermediate symbolic distance), which the metatrained network also reproduces (Fig. 3c,d).

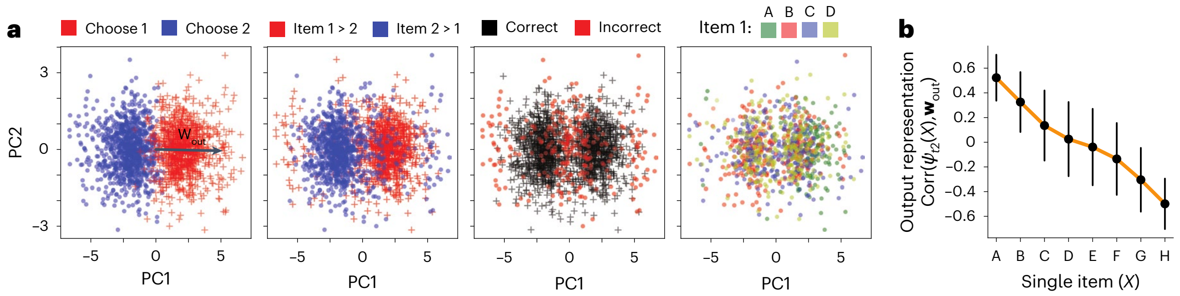


Fig. 4 | A simple representational scheme for transitive inference. Neural activity taken from trial 20 (first trial with nonadjacent pairs) across 2,000 separately run episodes (items generated independently for each episode). **a**, Principal component analysis (PCA) of neural activity. Each data point corresponds to $\mathbf{r}(t)$ at time step 2 of an individual trial. All plots show the same data, colored according to different trial type categories. Left, network response (choose item 1 vs. choose item 2); middle-left, correct choice (item 1 versus item 2); middle-right, correct versus incorrect; right, identity of item 1 (only four items shown for clarity). Note that PC1 is aligned with $\mathbf{w}_{\mathrm{out}}$ and effectively distinguishes between the different responses of network (left). **b**, Encoding of item rank. Alignment (correlation) of each item's step 2 neural representation $\boldsymbol{\psi}_{t2}(X)$ with the output weight vector $\mathbf{w}_{\mathrm{out}}$. Mean ± s.d. over episodes. Note the monotonic ordering (ranking) of items.

Neural mechanisms of transitive inference

A simple representational scheme for transitive inference

To understand the trained network's operation, we first examined network activity with principal component analysis (PCA). $\mathbf{r}(t)$ is the vector of neural activity at a given time (one element per neuron; Methods). We observed that the first principal component (PC1) of network activity $\mathbf{r}(t)$ at time step 2 of trial 20 (the last trial with adjacent-only pairs) is strongly aligned with the vector of output weights $\mathbf{w}_{\mathrm{out}}$, with correlation $r > 0.9$ (Fig. 4, left). As shown by the segregation of trials by the network's choice in Fig. 4a (red vs. blue points, left), position along this axis largely determines the network's choice response across trials. This is as expected because network response is read as the projection of $\mathbf{r}(t = 2)$ onto output weights (Methods).

Further, we found that $\mathbf{r}(t = 2)$ did not appear to contain information about the identity or rank of either individual item in the pair. Using various classification methods, we failed to reliably decode the rank of either the first or the second item from neural data at step 2 (Supplementary Fig. 5). Furthermore, neural activity vectors were not consistently separated by the rank of the first item (Fig. 4, right).

To understand how pairs of items are represented, we first examined the representations of isolated single items. We presented each stimulus $X \in$ (A, B, C, D, E, F, G, H) to the network as item 1 in isolation (not paired with any other) and observed the resulting neural activity $\mathbf{r}(t)$ at time steps 1 and 2. When showing item X in isolation, we denote $\mathbf{r}(t = 1)$ with the symbol $\boldsymbol{\psi}_{t1}(X)$ (the feedforward, step 1 representation of X, determined solely by the fixed, nonplastic input weights) and $\mathbf{r}(t = 2)$ with the symbol $\boldsymbol{\psi}_{t2}(X)$ (the learned, step 2 representation, produced by applying one step of recurrence through the learned, plastic weights to $\boldsymbol{\psi}_{t1}(X)$, and which determines the network's response for this trial).

We found that the plasticity-learned representation $\boldsymbol{\psi}_{t2}(X)$ of each stimulus X aligns with the network's decision axis (the output weight vector $\mathbf{w}_{\mathrm{out}}$) in proportion to item rank. That is, $\boldsymbol{\psi}_{t2}(A)$ has large positive correlation with $\mathbf{w}_{\mathrm{out}}$, $\boldsymbol{\psi}_{t2}(H)$ has large negative correlation with $\mathbf{w}_{\mathrm{out}}$ and intermediate items follow a monotonic progression (Fig. 4b).

In the actual task, stimuli are not presented in isolation, but in pairs. How are those learned representations of single items combined to represent pairs of items? Examining input weights W_{in}, we found that the input weights for the two items in the pair (that is, items 1 and 2; Fig. 1e) are strongly anticorrelated ($r \approx -0.9$). As a result, a given item's representation when shown as item 2 is essentially the negative of its representation when shown as item 1. Therefore, when a pair of items (X, Y) is presented as input, the network automatically computes a subtraction between the representations of both items ($\mathbf{r}(t = 1) \approx \boldsymbol{\psi}_{t1}(X) - \boldsymbol{\psi}_{t1}(Y)$). At the next time step, application of the recurrent weights approximately transforms this subtraction into $\boldsymbol{\psi}_{t2}(X) - \boldsymbol{\psi}_{t2}(Y)$ (neglecting the nonlinearity). Because $\boldsymbol{\psi}_{t2}(X)$'s alignment with the output weight vector $\mathbf{w}_{\mathrm{out}}$ is proportional to item rank, the alignment of $\mathbf{r}(t = 2) \approx \boldsymbol{\psi}_{t2}(X) - \boldsymbol{\psi}_{t2}(Y)$ with $\mathbf{w}_{\mathrm{out}}$, which determines the network's response, is proportional to the difference in rank between items X and Y.

This scheme provides a simple, intuitive mechanism for transitive inference—once the correct representation $\boldsymbol{\psi}_{t2}(X)$ for each individual item X has been learned, the subtractive operation immediately generalizes to nonadjacent pairs. Furthermore, more distant pairs imply a larger difference in the projection of either individual item along the decision axis; this suffices to cause a symbolic distance effect, where more distant item pairs are more accurately judged. A similar subtractive process has also been observed in models trained by abstract algorithms[13–15].

We note that the anticorrelated input weights for the two items in a stimulus pair were not specified by design, but emerged as a result of metatraining. We also note that these anticorrelated inputs are not strictly necessary for success—similar performance and abilities are obtained if W_{in} is frozen to its initial random values (not updated during metatraining) instead (Supplementary Note 6).

Representation learning

How does the metatrained network learn such order-encoding representations over the course of an episode? To investigate this how, we first assessed the dynamics of the neuromodulatory output $m(t)$ as this output is a simple, one-dimensional signal with a critical role in the network's learning (Methods). As shown in Fig. 5a,b, across trials, $m(t)$ at time step 3 (corresponding to the time of reward delivery for this trial) is consistently negative, regardless of reward received. By contrast, $m(t)$ at time step 4 (corresponding to the delay between trials) is strongly dependent on reward received at time step 3, being highly negative for negative rewards (incorrect trials) but positive or near zero for positive rewards (correct trials; Fig. 5c). This suggests that neuromodulatory outputs at time steps 3 and 4 of a given trial have different roles in learning.

We then examined changes in the plastic weights (and the resulting representations) at each successive time step of a trial. Reasoning that error trials would be most useful in clarifying how the network learns, we selected runs in which the network sees the pair DE or ED at trial 20, but not in any previous trial (we focus on DE for illustration only; similar results hold for other pairs; Extended Data Fig. 2). This presentation of a new pair of items with intermediate ranks, not previously encountered, invariably leads to an erroneous response in this trial. At this stage, the network has only encountered D and E as the lowest- and highest-ranked items, respectively, within two separate sublists (A–D and E–H) misrepresenting their true order, D > E. We sought to characterize the actual representational changes resulting from this error.

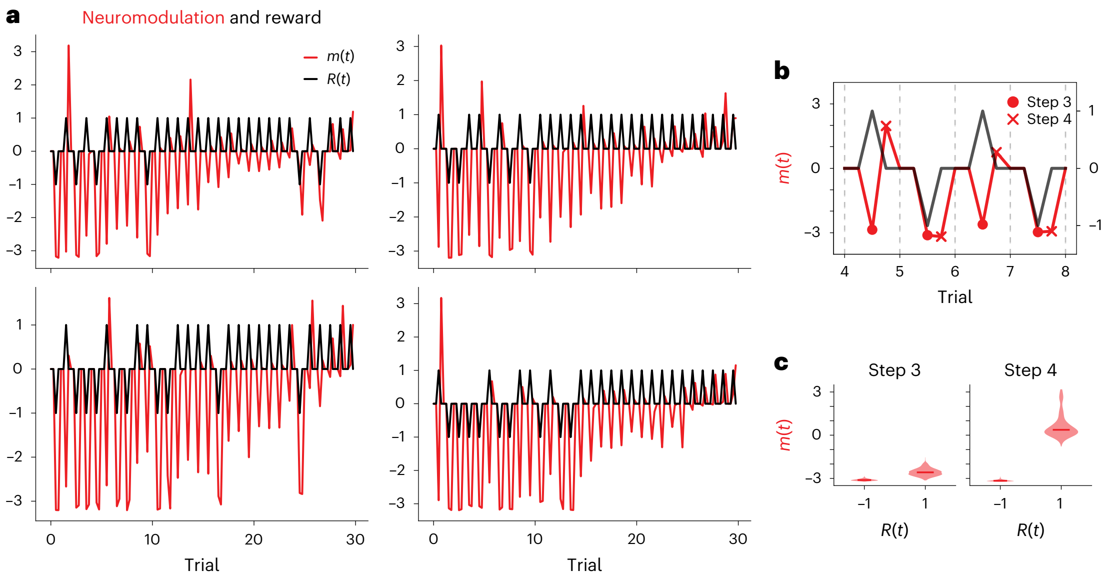


Fig. 5 | Two distinct patterns of neuromodulation. **a**, Neuromodulatory output $m(t)$ and reward signal $R(t)$ for four example episodes (each episode having independently generated items). For clarity, $m(t)$ is plotted as 0 for the first two time steps of each trial, when its value is irrelevant ('Main' and Methods). Each 'spike' in the black (reward) curve corresponds to the reward delivered at step $t = 3$ of one trial (positive or negative). **b**, Expanded view of trials 4–8 from the top-right example in **a**. Note how $m(t = 3)$ (red circle) is consistently negative independently of reward sign, while $m(t = 4)$ (red cross) is negative for negative rewards and positive or near 0 for positive reward. **c**, Violin plot of $m(t = 3)$ and $m(t = 4)$ for an early trial (trial 5) across 2000 separately run episodes, shown separately for correct (reward = +1) versus incorrect (reward = −1) responses.

In particular, we computed the network's learned (step 2) representations of each isolated item $\boldsymbol{\psi}_{t2}(X)$. Moreover, we computed this representation across time in the trial, using frozen plastic weights $P(t)$ extracted at each of the four successive time steps (of trial 20). This allowed us to observe $\boldsymbol{\psi}_{t2}(X)$ as it was represented by the network at each specific time step of the trial. Then, we computed the correlation of each of these successive representations with the output weight vector $\mathbf{w}_{\text{out}}$, as in Fig. 4b, thereby enabling us to observe how the network's estimated rank of item X changes across time steps as a result of plasticity (Fig. 6, top row).

We found that plasticity in step 3 (when the reward signal is delivered to the network) produces small weight changes restricted to items D and E (Fig. 6, bottom row, third column). By contrast, step 4 produces not only relatively larger changes for items D and E, but, crucially, a substantial change (of appropriate sign) for items C and F (Fig. 6, bottom row, fourth column), neighboring items not presented in the trial. This extension of representation changes to additional items serves to 'reassemble' previously separate orderings into a single global ordering (Fig. 6, top row, compare right (step 4) and middle-right (step 3)), consistent with the network's ability to perform list linking.

Reinstatement of recoded representations supports learning

How do these plastic changes occur mechanistically? In the model, plasticity at time step k can only access activity over the preceding two time steps (equations (5) and (7)). This appears to pose a problem, as stimuli are presented at time step 1, yet the crucial plasticity (as shown above) occurs at time step 4. This problem implies that the network can somehow reinstate representations of relevant stimuli (including nonpresented items) at time step 2, thus enabling the appropriate plasticity to take place at time step 4.

However, in initial analyses, we failed to find reinstatement of original (feedforward) stimulus representations. Feedforward representations $\boldsymbol{\psi}_{t1}(X)$ were strongly present in neural activity at time step 1 (that is, upon stimulus presentation), as expected, but were not found at time step 2 or any other time step (Fig. 7a). Thus, the delayed learning at time step 4 observed in Fig. 6 did not involve reactivation (or persistence) of the original stimulus representations.

Could this lack of reactivated representation reflect a different kind of representation? We considered that the network is not uniformly plastic—each connection has a different baseline nonplastic weight $W_{i,j}$ and a different plasticity coefficient $A_{i,j}$. Therefore, to produce the appropriate synaptic changes through Hebbian plasticity, the relevant reinstated representations should not be identical to the original (feedforward) representation of each item $\boldsymbol{\psi}_{t1}(X)$. Rather, such plasticity would involve recoded versions of these representations, which would produce appropriate learning (that is, ensure that Hebbian learning adjusts learned representations of the relevant items in the correct direction along the decision axis) when taking into account the heterogeneous plasticity across individual synapses.

To identify these putative recoded representations, we designed an optimization-based procedure that captures the above hypotheses (Supplementary Note 8). This procedure yielded a predicted recoded version of the feedforward representations ($\boldsymbol{\psi}_{t1}(X)$), which we termed $\tilde{\boldsymbol{\psi}}_{t1}(X)$, and of the decision axis ($\mathbf{w}_{\text{out}}$), which we termed $\tilde{\mathbf{w}}_{\text{out}}$.

We found that the relevant items are indeed represented in neural activity, in the predicted recoded forms $\tilde{\boldsymbol{\psi}}_{t1}(X)$, precisely at time step 2 and with appropriate signs. Figure 7b shows the correlation between $\mathbf{r}(t)$ and the predicted recoded feedforward representations $\tilde{\boldsymbol{\psi}}_{t1}(X)$ of all items, for each step of trial 20, again using only runs in which the pair shown at trial 20 was either DE or ED. Critically, in addition to current-trial items (D and E), neighboring items (C and F) are also reinstated. Furthermore, we found that the recoded representation of the decision axis (output weight vector) is represented specifically at time step 3, with the same sign as the network's response for this trial (Extended Data Fig. 3 and data for other pairs is shown in Fig. 8). Together with the error-selective $m(t)$ signal at $t = 4$, and neuromodulated Hebbian learning as governed by equations (5) and (7), this suffices to explain the shift in learned step 2 representations of the above items in the appropriate direction, as observed in Fig. 6 (Supplementary Note 1).

Importantly, the recoded representations $\tilde{\boldsymbol{\psi}}_{t1}(X)$ are markedly different from the original representations evoked by the actual stimuli $\boldsymbol{\psi}_{t1}(X)$. Correlation between $\tilde{\boldsymbol{\psi}}_{t1}(X)$ and $\boldsymbol{\psi}_{t1}(X)$ for any item X was consistently below 0.1 in magnitude, and could be of either sign. This clarifies

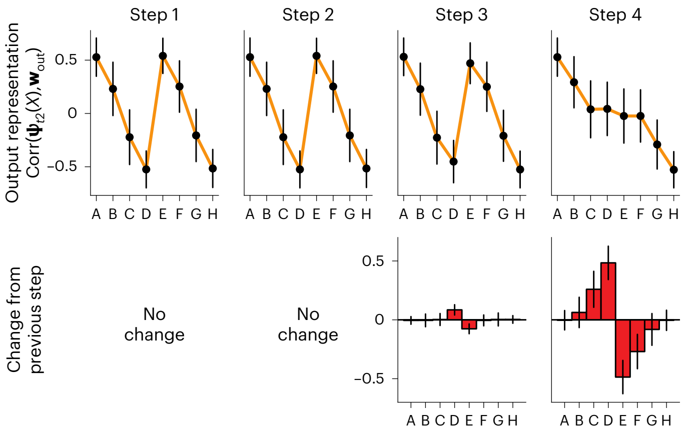


Fig. 6 | Presentation of an item pair alters representations of nonpresented neighboring items. Activity taken from network when presented with pair DE or ED for the first time at trial 20, over 1000 separately run episodes. Top row, encoding of items across time steps. Plotted is the alignment (correlation) of each item's step 2 (learned) representation $\psi_{t2}(X)$ with the output weight vector $\mathbf{w}_{out}$. Bottom row, changes in alignment from one time step to the next (that is, difference between each top-row plot over time steps). Plots show mean ± s.d. over episodes. Note that most learning occurs in step 4 (delay step); further, note that learning affects the representation of not only the pair presented in the trial (DE), but also neighboring items that were not shown in the trial (C and F). No weight changes occur in steps 1 and 2 due to reset of activations at the start of trials (Methods).

why we did not find reactivation of the original stimulus representations (Fig. 7a). Therefore, reinstated representations that support delayed learning in the network are not generally similar to the original feedforward representations, although they contain the appropriate information to enable learning for the corresponding items (Discussion and Supplementary Note 8).

Joint reinstatement of adjacent stimuli requires the model to identify which items are adjacent to one another. How is this learned? We found that the relevant learning occurs at time step 3, corresponding to the small, reward-independent spike in $m(t = 3)$ previously seen (Fig. 5). This modulatory spike induces Hebbian learning between the feedforward representations of item pairs presented at time step 1, and these pairs' reinstated, recoded representations at time step 2. The result is that future presentations of either item in the pair will induce reinstatement of the other item. Since stimuli up to trial 20 are always comprised of pairs of adjacent items, this learning mechanism explains joint reinstatement of adjacent stimuli (Fig. 7b). Importantly, the consistently negative sign of $m(t = 3)$ ensures that this joint reinstatement occurs with a common sign. See Supplementary Note 4 for details and additional experiments.

A more precise step-by-step summary of the discovered neural algorithm is provided in Supplementary Note 1. We also evaluated a non-neural, toy-model implementation of the algorithm, verifying that this overall algorithm suffices to produce appropriate representations within an episode (Supplementary Note 2).

A suboptimal solution cannot perform list linking

As mentioned earlier ('Metalearning discovers solutions for transitive inference'), the metatraining process sometimes yields networks that perform at a substantially lower level (Fig. 1f, gray traces), suggesting that these networks implement a different learning method. Indeed, analysis of these networks revealed an alternative solution to order learning, based on simple reward-modulated Hebbian learning: neuromodulatory output $m(t)$ was strongly sensitive to reward at $t = 3$ (reward delivery time; Supplementary Fig. 9). As denoted in equation (7), this signal automatically modulates Hebbian learning between feedforward item input (at step 1) and response (at step 2). Representation learning only appears to occur at time step 3 (reward delivery time; Supplementary Figs. 10 and 11). Finally, we note that after the first several trials, step 3 learning in these networks only occurred in incorrect trials (negative rewards).

This 'passive', nonreinstating Hebbian solution was capable of passable transitive inference, yet, tellingly, failed to perform list linking (Extended Data Fig. 1), unlike the 'active', cognitive solution described previously. As the passive solution only involves current-trial information, it is unable to rapidly modify representations of items not currently presented (knowledge 'reassembly'). See Supplementary Note 11 for details.

Discussion

In this study, we metatrained neural networks, endowed with synaptic plasticity and neuromodulation, in a classical transitive inference task paradigm. The metatraining process consistently produced networks that were capable of learning arbitrary orderings for transitive inference, and, further, that reproduced multiple important experimentally observed behaviors (symbolic distance effect, end-anchor effect, cross-list rank transfer, resistance to massed presentations). To our knowledge, these networks are the only neural models described thus far to do so[11].

Surprisingly, despite the simplicity of the networks, the learning mechanism employed (as discovered by the metalearning process) relied on a cognitive act—relevant previous stimuli were actively and selectively reinstated in neural activity (in recoded form), enabling trial-delayed (temporally remote) changes in their representations.

As a result of this reinstatement, and despite not being explicitly trained to do so, this solution was able to perform list linking, a task paradigm of fast knowledge reassembly[10] (observed in both monkeys[8,9,24] and humans[10]), moreover reproducing experimentally observed patterns of errors (Fig. 3 and ref. 9).

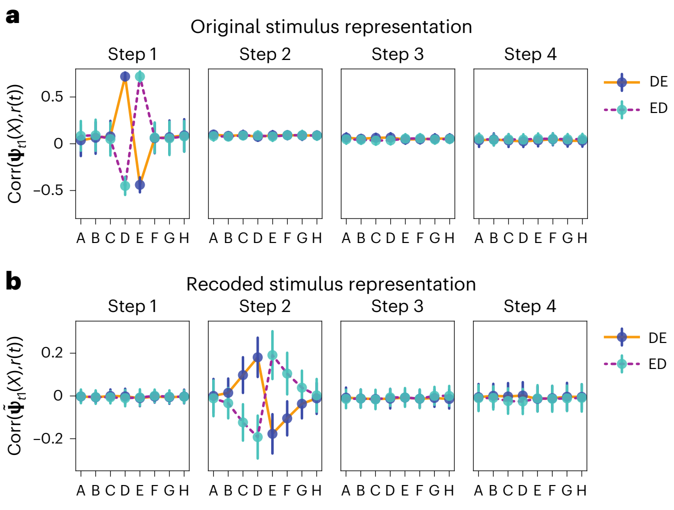


Fig. 7 | Metatrained networks reinstate item representations in a recoded form. Activity taken from network when presented with pair DE or ED for the first time at trial 20, over separately run episodes (DE, 136 episodes and ED, 142 episodes). **a**, Original stimulus representation. Alignment (correlation) between neural activity $\mathbf{r}(t)$ and stimulus-evoked representations of individual stimuli (ψ_{t1}), across items and time steps. **b**, Recoded stimulus representation. Same calculation as **a**, but using $\tilde{\psi}_{t1}$ (putative recoded representations found via optimization; Supplementary Note 8). The network expresses recoded representations of D and E at time step 2, doing so with opposite sign depending on whether the item is item 1 or item 2. Critically, neighboring items (C and F, and less strongly B and G) are also reinstated in recoded form and with appropriate signs. All plots indicate mean ± s.d. over episodes.

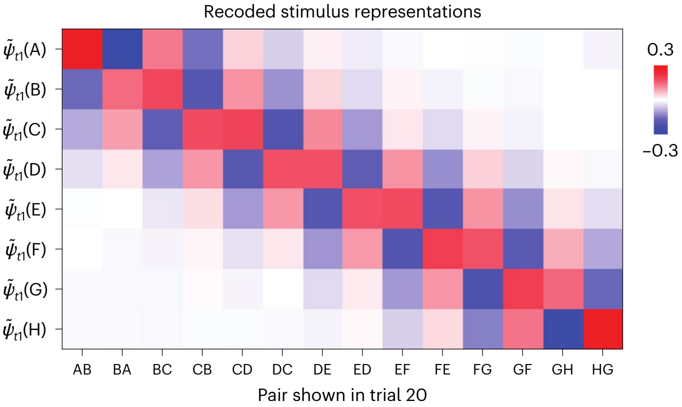


Fig. 8 | Reinstating recoded stimulus representations—additional item pairs. Alignment (correlation) between $\mathbf{r}(t)$ and $\tilde{\boldsymbol{\Psi}}_{t1}(X)$ for all items at time step 2 of trial 20. Values were averaged across all 2,000 separately run episodes. Note that data for DE and ED are the same as in Fig. 7b for time step 2.

By contrast, a minority of the resulting networks learned order through simple reward-modulated Hebbian learning, applied to immediate stimuli, responses and rewards. This 'passive', nonreinstating solution was still capable of passable transitive inference at test time, suggesting that transitive inference stricto sensu (which James once called 'the broadest and deepest law of man's thought'[25] (p. 646)) does not require advanced cognition[15]. Importantly, however, this nonreinstating solution was not capable of performing list linking (Supplementary Fig. 9). This striking difference between the metatrained models echoes differences in list-linking (and order learning) ability across animal species[6,8] and human participants[10], and, further, is apropos to the inability of classical learning models to perform list linking[8,11].

The reinstatement found in the dominant solution is reminiscent of neural activity that appears to reflect reactivation of memory traces, such as hippocampal 'replay'[26–28], which has been suggested as a component of relational learning[28–32]. However, traditional replay tends to occur spontaneously and also often outside of periods of overt task engagement[26,27,30]. By contrast, reinstatement in our model is evoked by stimulus presentation at fixed time points in a trial, and is directly conjoined to representation learning within a given trial. Such stimulus-evoked, learning-period reactivation is more broadly consistent with the finding of 'retrieval-mediated learning'[33,34], which has been linked to relational learning in the associative inference task paradigm[29,35]. Our results suggest that such reactivation can also have a role in order learning paradigms, not directly for response production, but rather to modify previously established representations in accordance with relational structure (knowledge 'reassembly').

We also found that representations of previous stimuli were reinstated in a recoded form. Reactivation is often measured by testing for the re-occurrence of patterns similar to those evoked by the original stimuli[30,31], or by computing the similarity between neural activity during and after encoding[28]. Our results emphasize that reinstated representations need not be identical, or even similar, to those directly elicited by stimulus presentation, but may differ from them seemingly arbitrarily—as long as they produce the desired effect on representations (Fig. 7, compare plots, and Supplementary Note 8).

The 'active', reinstating solution uses reinstatement for learning, rather than for inference at decision time. By design, agent responses (produced at step 2, after exactly one pass through the plastic recurrent weights) cannot involve complex processing, but must operate as a model-free 'learned reflex'. By contrast, the delayed, self-generated learning at step 4 is model-based: it relies on reinstatement, and the agent must know which items neighbor each other (a simple model of its environment) to reinstate the correct ones. Therefore, the active, reinstatement-based solution combines model-based learning with model-free inference, reminiscent of the so-called Dreamer algorithm from the machine learning literature[36].

Finally, our results illustrate the potential of modeling approaches based on plastic artificial neural networks. Many recent studies have trained recurrent neural networks with backpropagation on various tasks, identifying rich dynamics that support the performance of these tasks[14,37–39]. The present study extends this approach to plastic networks, capable of controlling their own learning, and metatrained (using gradient descent) to discover new learning processes that can autonomously perform cognitive learning tasks. Our results demonstrate the power of this approach for investigating the space of neural mechanisms that support learning and cognition.

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/s41593-024-01852-8.

References

1. 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).
2. Tervo, D. G. R., Tenenbaum, J. B. & Gershman, S. J. Toward the neural implementation of structure learning. *Curr. Opin. Neurobiol.* **37**, 99–105 (2016).
3. Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behav. Brain Sci.* **40**, e253 (2017).
4. Eichenbaum, H. Memory: organization and control. *Annu. Rev. Psychol.* **68**, 19–45 (2017).
5. Behrens, T. E. J. et al. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron* **100**, 490–509 (2018).
6. Jensen, G. Serial learning. In *APA Handbook of Comparative Psychology: Perception, Learning, and Cognition* (ed. Call, J.) 385–409 (American Psychological Association, 2017).
7. Gazes, R. P., Templer, V. L. & Lazareva, O. F. Thinking about order: a review of common processing of magnitude and learned orders in animals. *Anim. Cogn.* **26**, 299–317 (2023).
8. Gazes, R. P. & Lazareva, O. F. Does cognition differ across species, and how do we know? Lessons from research in transitive inference. *J. Exp. Psychol. Anim. Learn. Cogn.* **47**, 223 (2021).
9. Treichler, F. R. & van Tilburg, D. Concurrent conditional discrimination tests of transitive inference by macaque monkeys: list linking. *J. Exp. Psychol. Anim. Behav. Process.* **22**, 105 (1996).
10. Nelli, S., Braun, L., Dumbalska, T., Saxe, A. & Summerfield, C. Neural knowledge assembly in humans and neural networks. *Neuron* **111**, 1504–1516 (2023).
11. Jensen, G., Terrace, H. S. & Ferrera, V. P. Discovering implied serial order through model-free and model-based learning. *Front. Neurosci.* **13**, 878 (2019).
12. Morton, N. W., Schlichting, M. L. & Preston, A. R. Representations of common event structure in medial temporal lobe and frontoparietal cortex support efficient inference. *Proc. Natl Acad. Sci. USA* **117**, 29338–29345 (2020).
13. De Lillo, C., Floreano, D. & Antinucci, F. Transitive choices by a simple, fully connected, backpropagation neural network: implications for the comparative study of transitive inference. *Anim. Cogn.* **4**, 61–68 (2001).
14. Kay, K. et al. Emergent neural dynamics and geometry for generalization in a transitive inference task. *PLOS Comput. Biol.* **20**, e1011954 (2024).

15. Lippl, S., Kay, K., Jensen, G., Ferrera, V. P. & Abbott, L. F. A mathematical theory of relational generalization in transitive inference. *Proc. Natl Acad. Sci. USA* **121**, e2314511121 (2024).
16. Schmidhuber, J. Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook (diploma thesis). www.bibsonomy.org/bibtex/2a96f7c3d42103ab94b13badef5d869f0/brazovayeye (1987).
17. Miconi, T. Backpropagation of Hebbian plasticity for continual learning. In *Proceedings of NIPS Workshop on Continual Learning* (2016).
18. Wang, J. X. et al. Learning to reinforcement learn. Preprint at https://doi.org/10.48550/arXiv.1611.05763 (2016).
19. Wang, J. X. et al. Prefrontal cortex as a meta-reinforcement learning system. *Nat. Neurosci.* **21**, 860 (2018).
20. Miconi, T., Rawal, A., Clune, J. & Stanley, K. O. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In *Proceedings of International Conference on Learning Representations* (2019).
21. Mnih, V. et al. Asynchronous methods for deep reinforcement learning. In *Proceedings of International Conference on Machine Learning* (eds Balcan, M. F. & Weinberger K. Q.) Vol. 48, 1928–1937 (PMLR, 2016).
22. Brunamonti, E. et al. Neuronal modulation in the prefrontal cortex in a transitive inference task: evidence of neuronal correlates of mental schema management. *J. Neurosci.* **36**, 1223–1236 (2016).
23. Lazareva, O. F. & Wasserman, E. A. Transitive inference in pigeons: measuring the associative values of stimuli B and D. *Behav. Process.* **89**, 244–255 (2012).
24. Treichler, F. R., Raghanti, M. A. & van Tilburg, D. N. Linking of serially ordered lists by macaque monkeys (*Macaca mulatta*): list position influences. *J. Exp. Psychol. Anim. Behav. Process.* **29**, 211 (2003).
25. James, W. *The Principles of Psychology* Vol. 2. (Henry Holt and Company, 1890).
26. Buzsáki, G. Hippocampal sharp wave-ripple: a cognitive biomarker for episodic memory and planning. *Hippocampus* **25**, 1073–1188 (2015).
27. Foster, D. J. Replay comes of age. *Annu. Rev. Neurosci.* **40**, 581–602 (2017).
28. Tambini, A. & Davachi, L. Awake reactivation of prior experiences consolidates memories and biases cognition. *Trends Cogn. Sci.* **23**, 876–890 (2019).
29. Shohamy, D. & Daw, N. D. Integrating memories to guide decisions. *Curr. Opin. Behav. Sci.* **5**, 85–90 (2015).
30. Barron, H. C. et al. Neuronal computation underlying inferential reasoning in humans and mice. *Cell* **183**, 228–243 (2020).
31. Comrie, A. E., Frank, L. M. & Kay, K. Imagination as a fundamental function of the hippocampus. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **377**, 20210336 (2022).
32. Kurth-Nelson, Z. et al. Replay and compositional computation. *Neuron* **111**, 454–469 (2023).
33. Iordanova, M. D., Good, M. & Honey, R. C. Retrieval-mediated learning involving episodes requires synaptic plasticity in the hippocampus. *J. Neurosci.* **31**, 7156–7162 (2011).
34. Hall, G. Learning about associatively activated stimulus representations: implications for acquired equivalence and perceptual learning. *Anim. Learn. Behav.* **24**, 233–255 (1996).
35. Zeithamova, D., Dominick, A. L. & Preston, A. R. Hippocampal and ventral medial prefrontal activation during retrieval-mediated learning supports novel inference. *Neuron* **75**, 168–179 (2012).
36. Hafner, D., Lillicrap, T., Ba, J. & Norouzi, M. Dream to control: learning behaviors by latent imagination. Preprint at https://doi.org/10.48550/arXiv.1912.01603 (2019).
37. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
38. Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T. & Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nat. Neurosci.* **22**, 297–306 (2019).
39. Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F. & Lin, H. (eds). Organizing recurrent network dynamics by task-computation to enable continual learning. In *Proceedings of Advances in Neural Information Processing Systems* Vol. 33, 14387–14397 (Curran Associates, 2020).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

Terminology

Metalearning, or 'learning-to-learn', consists of training an agent so that it can solve new instances of a general learning problem[16,18,19,40]. That is, instead of training the agent separately on each new instance of the problem, we metatrain an autonomous learning agent so that it acquires the ability to learn autonomously and efficiently any given instance of the problem, including new instances never seen during training. Classic work by Harlow[41] first showed that animals possess this ability in an item-value association metatask—over the course of multiple episodes, monkeys became progressively better at learning (and exploiting) which of two possible items was associated with a food reward. Typical metalearning problems in the literature include maze solving[42,43], bandit tasks[18,42], fast association between stimuli and labels[18,43,44] and item-value problems such as the Harlow metatask[19,41]. Here we apply a metalearning framework to plastic neural networks, training them to be able to learn new ordered series of arbitrary stimuli from pairwise presentations, as in classic transitive inference experiments[8,11,45].

In the following, to avoid confusion, we will maintain separate usage for the words learning and training. We use the word 'learning' to denote within-episode learning of one particular ordered list, driven by synaptic plasticity in plastic weights. In contrast, we use the word 'training' to denote the gradient-based optimization that occurs between episodes, and modifies the structural parameters of the network, with the goal to improve within-episode plasticity-driven learning. This distinction follows that between 'inner loop' and 'outer loop', respectively, in standard metalearning terminology[18,42].

In each trial, after the network produces a response, it is given a binary feedback signal $R(t)$ that indicates whether this response was correct or not. We call this signal 'reward', although we stress that it is merely an additional input to the network, and has no predefined effect on network structure by itself.

Finally, we use 'stimulus' and 'item' interchangeably.

Network organization

The network is a recurrent neural network endowed with Hebbian plasticity and self-controlled neuromodulation, containing $n = 200$ neurons (Fig. 1). The general structure of the network and learning process largely follows those of a previous study[20]. The code was written entirely in PyTorch and run on Google Colab. All code for the following experiments is available online (Code availability).

The inputs $\mathbf{i}(t)$ consist of a vector that concatenates the stimuli for the current time step, the reward signal $R(t)$ for the current time step (0, 1 or −1), and the agent's response at the previous time step (if any), in accordance with common metalearning practice[18,19,42]. The network's output is a probability distribution over the two possible responses ('choose stimulus 1' or 'choose stimulus 2'). At response time, the agent's actual response for this trial is sampled from this output distribution; at other times, the outputs are ignored.

Recurrent connections within the network undergo synaptic changes over the course of an episode, modeled as simple neuromodulated Hebbian plasticity as described in the 'Synaptic plasticity' section.

Formally, the fully connected recurrent network acts according to the following equations:

$$\mathbf{x}(t) = W_{\text{in}}\,\mathbf{i}(t) + (W_{\text{rec}} + A \odot P(t))\,\mathbf{r}(t-1) \quad (1)$$

$$\mathbf{r}(t) = \tanh(\mathbf{x}(t)) \quad (2)$$

$$o(t) = \text{Softmax}(W_{\text{out}}\,\mathbf{r}(t)) \quad (3)$$

$$m(t) = W_{\text{mod}}\,\mathbf{r}(t) \quad (4)$$

Here $\mathbf{i}(t)$ is the vector of inputs provided to the network, $\mathbf{x}(t)$ is the vector of neural activations (the linear product of inputs by weights), $\mathbf{r}(t)$ is the neural firing rates (activations passed through a nonlinearity), $o(t)$ is the output of the network (that is, the probability distribution over the two possible responses) and $m(t)$ is the (scalar-valued) neuromodulatory output whose role is to gate synaptic changes (plasticity; see below).

W_{rec} and A are the base weights and plasticity parameters (plasticity learning rates) of the recurrent connections, respectively. They are structural parameters of the network, and do not change during an episode, but rather are slowly optimized between episodes by gradient descent. By contrast, $P(t)$ is the plastic component of the weights, reset to 0 at the start of each episode and changing over an episode according to the plasticity rule described below. The symbol $\odot$ denotes the pointwise (Hadamard) product of two matrices.

We note that these equations are simply the standard recurrent neural network equations, except that the total recurrent weights are the sum of base weights W_{rec} and the plastic weight term $A \odot P(t)$.

Crucially, within an episode, only the plastic recurrent weights $P(t)$ are updated, with the updating governed by the plasticity rule described below. All other parameters (W_{in}, W_{rec}, W_{out}, W_{mod} and A) are fixed and unchanging within an episode, but are optimized by gradient descent between episodes.

In all experiments, the final trained output weight matrix W_{out} were found to have two highly anticorrelated rows ($r < 0.9$), corresponding to the two opposite possible decisions (choose item 1 vs. choose item 2). Thus, in the text, we generally summarize the W_{out} matrix by a single vector $\mathbf{w}_{\text{out}}$, computed as the difference between the two rows of W_{out}. All results are unchanged (up to sign) if $\mathbf{w}_{\text{out}}$ was instead taken to be either row of W_{out}.

Synaptic plasticity

To model synaptic learning, recurrent connections are endowed with modulable Hebbian plasticity[20]. Each connection maintains a so-called Hebbian eligibility trace $H(t)$, which is a decaying running average of the product of outputs and inputs.

$$H(t+1) = \eta\,\mathbf{x}(t)\,\mathbf{r}(t-1)^{\mathrm{T}} + (1-\eta)H(t) \quad (5)$$

Since the network is recurrent, $\mathbf{x}(t)$ corresponds to the 'outputs', while $\mathbf{r}(t-1)$ corresponds to the 'inputs' across the recurrent connections (cf. equation (1)).

Finally, the network continually produces a neuromodulation signal $m(t)$ that gates the Hebbian trace into actual changes in plastic weights P.

$$P(t+1) = P(t) + m(t)\,H(t) \quad (6)$$

We emphasize that $P(t)$ is initialized to 0 at the beginning of each episode, and changes according to the above equations (without reinitialization) over the course of an episode. Also note that while $H(t)$ is decaying, $P(t)$ does not decay within an episode.

In equation 5, η is a network-wide parameter, optimized by gradient descent between episodes. Notably, across many experiments, we observed that training consistently settles on a value of $\eta \approx 1$. This means that $H(t)$ is almost fully updated at each time step, with essentially no memory of events occurring before time $t-2$. As such, we can approximate the actual plasticity process as follows:

$$P(t+1) \approx P(t) + m(t)\,\mathbf{x}(t-1)\,\mathbf{r}(t-2)^{\mathrm{T}} \quad (7)$$

This observation has an important role in the elucidation of network behavior, as described in the main text.

Task settings

Our objective is not simply to train successful learning networks but also to fully understand the neural mechanism of learning in these trained networks. To this end, we deliberately sought to simplify trial structure as much as possible while retaining the essential structure of real-world experiments.

First, we restricted each trial to the shortest possible duration that still allowed for successful training, which we found to be exactly four time steps—stimulus presentation at step 1, network response at step 2, external feedback (reward) at step 3 and, lastly, a delay time step before the start of the next trial, as step 4.

Second, we reset neural activations $\mathbf{x}(t)$, $\mathbf{r}(t)$ and Hebbian eligibility traces $H(t)$ at the start of each trial (but not plastic weights $P(t)$). This is to ensure that neural activity dynamics remain confined to a single trial, with only plastic weights carrying memory of past events from one trial to the next. If we do not reset neural activations between trials, trained networks consistently settle on strategies that process information from successive trials in parallel; that is, activations in trial k represent information not just from trial k but also from trial $k-1$. While potentially interesting, this phenomenon complicates the investigation of mechanisms underlying within-episode learning. We therefore chose to reset activations at the start of each trial.

Rewards in the last ten episodes are upweighted by a factor of 4; this only modifies the metalearning loss for outer-loop gradient descent and does not affect the reward signal registered by the network.

Reporting summary

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

Code availability

All code for the experiments described above, as well as saved parameter files for representative networks for both discovered solutions, is available at https://github.com/ThomasMiconi/TransitiveInference.

References

40. Thrun, S. & Pratt, L. (eds). *Learning to Learn: Introduction and Overview* 3–17 (Kluwer Academic Publishers, 1998).
41. Harlow, H. F. The formation of learning sets. *Psychol. Rev.* **56**, 51–65 (1949).
42. Duan, Y. et al. RL^2: fast reinforcement learning via slow reinforcement learning. Preprint at https://doi.org/10.48550/arXiv.1611.02779 (2016).
43. Miconi, T., Clune, J. & Stanley, K. O. Differentiable plasticity: training plastic networks with gradient descent. In *Proc. 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) Vol. 80, 3559–3568 (PMLR, 2018).
44. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D. & Lillicrap, T. Meta-learning with memory-augmented neural networks. In *Proc. 33rd International Conference on Machine Learning* (eds Balcan, M. F. & Weinberger, K. Q.) Vol. 48, 1842–1850 (PMLR, 2016).
45. Bryant, P. E. & Trabasso, T. Transitive inferences and memory in young children. *Nature* **232**, 456–458 (1971).

Acknowledgements

We thank L. Abbott, V. P. Ferrera, S. Lippl, G. Jensen and X. X. Wei for fruitful discussions about this work and comments on an earlier version of the paper. This work was supported by the Simons Collaboration for the Global Brain (521921 and 542981), the NSF NeuroNex award (DBI-1707398), the Gatsby Charitable Foundation and NIMH K99 (MH126158-01A1 to K.K.).

Author contributions

T.M. designed and ran the code. T.M. and K.K. conceived the experiment and wrote the paper.

Competing interests

The authors declare no competing interests.

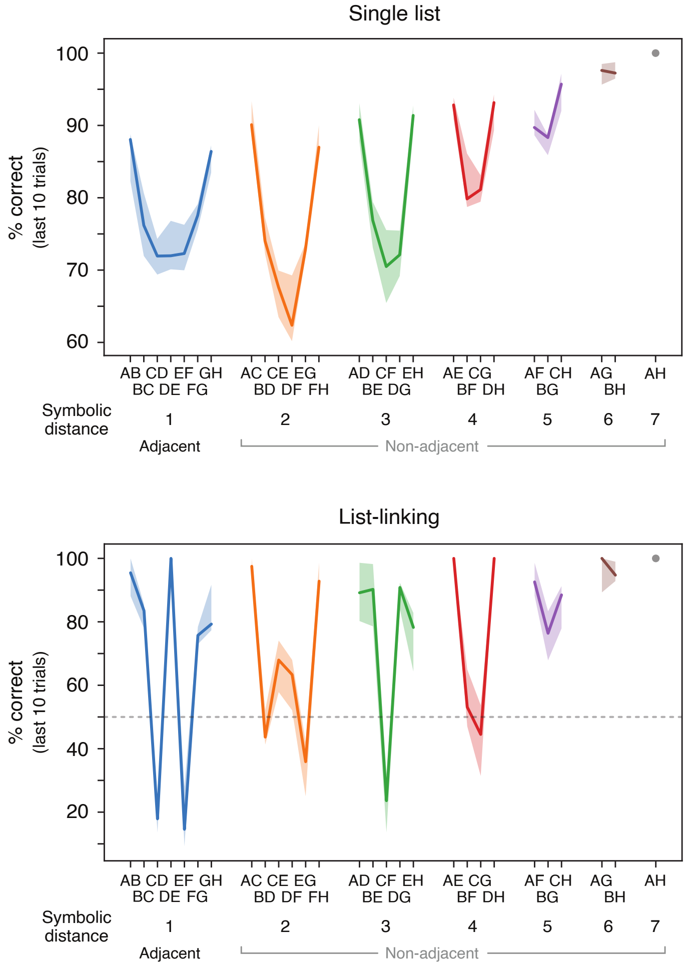
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s41593-024-01852-8.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41593-024-01852-8.

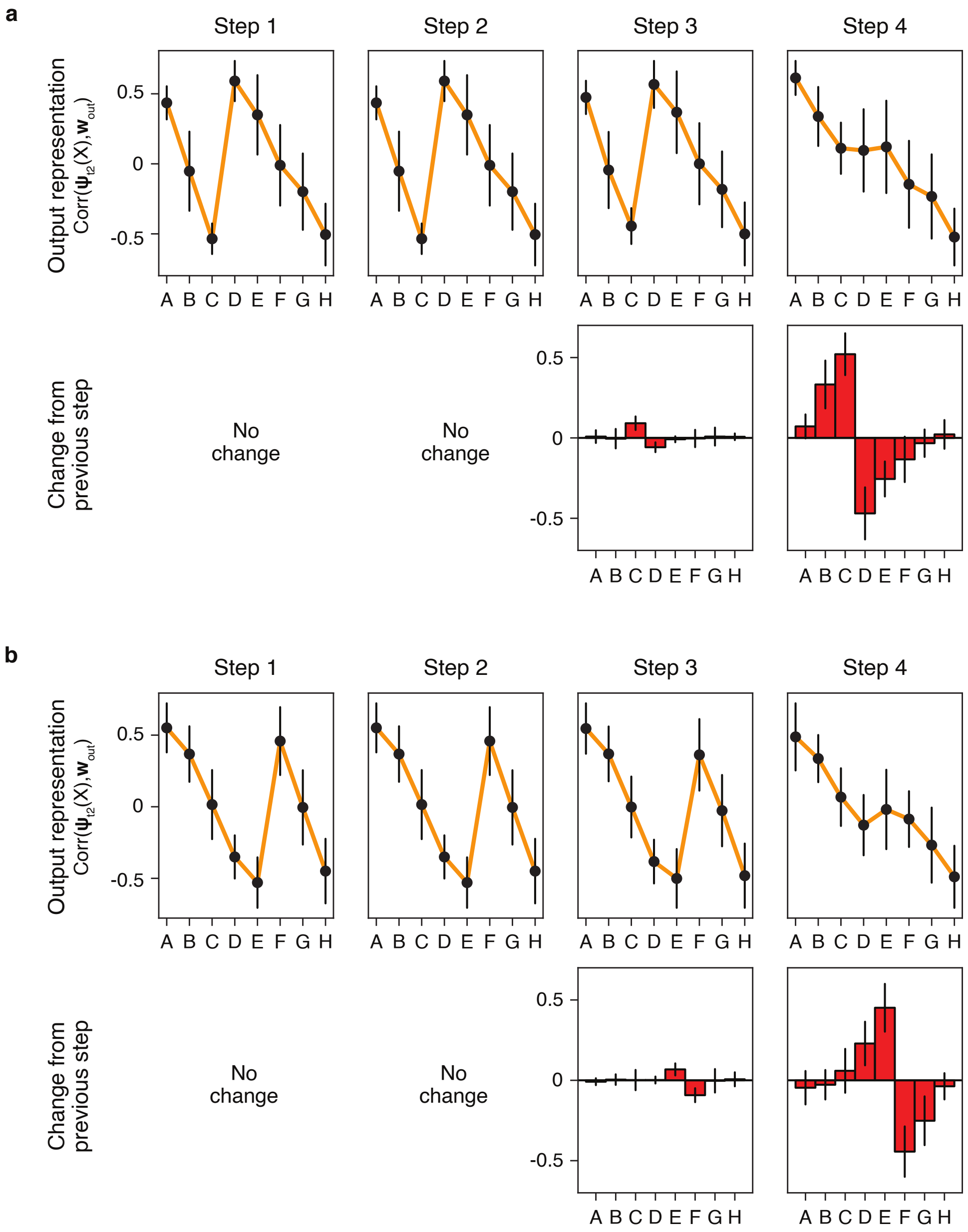
Correspondence and requests for materials should be addressed to Thomas Miconi or Kenneth Kay.

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



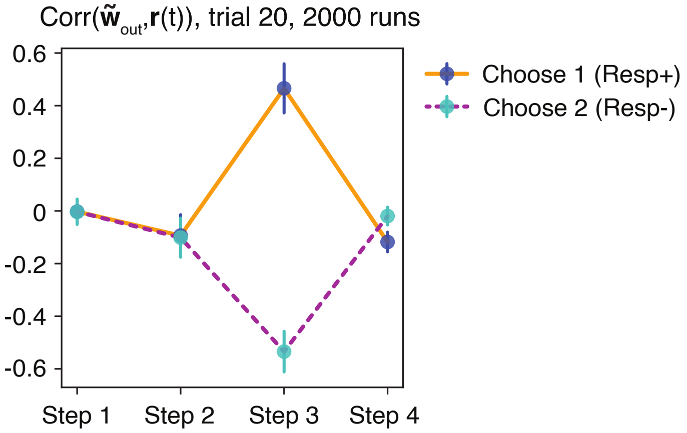
Extended Data Fig. 1 | Performance of the suboptimal network. Proportion of correct test trials, in standard (single-list) conditions (top; compare to high-performance network in Fig. 2a) and list-linking conditions (bottom; compare to high-performance network in Fig. 3a), over 2,000 episodes. Median and interquartile range of correct responses for each item pair over 10 sets of 200 episodes each. Note that the suboptimal network performs transitive inference and expresses the symbolic distance effect for a single list, but fails the list-linking task. Conventions as in Figs. 2 and 3.

https://doi.org/10.1038/s41593-024-01852-8



Extended Data Fig. 2 | Step-by-step representation changes for different withheld pairs. Conventions are as in Fig. 6 (showing mean ± s.d. over 1000 separately run episodes), but here the pair that is withheld until trial 20 is either CD (**a**) or EF (**b**) rather than DE. Representation learning proceeds similarly for these other pairs.

https://doi.org/10.1038/s41593-024-01852-8



Extended Data Fig. 3 | Reinstatement of recoded decision axis. Correlation between neural activity r(t) and recoded vector of output weights $\tilde{\mathbf{w}}_{out}$, at each time step of trial 20 (mean ± s.d. over 2000 separately run episodes), shown separately for runs in which network response at trial 20 was positive vs. negative. While actual response occurs at step 2, the recoded representation of the combined output weight vector $\tilde{\mathbf{w}}_{out}$ is strongly represented at step 3, with sign correlated with network response for this trial. This representation enables Hebbian association with recoded representations of stimuli reinstated at step 2 (Fig. 6).