Active reconfiguration of neural task states

Harrison Ritz[1,✉], **Aditi Jha**[1,3], **Jonathan Pillow**[1], **Nathaniel D. Daw**[1,2], **and Jonathan D. Cohen**[1,2]

[1]Princeton Neuroscience Institute, Princeton University
[2]Department of Psychology, Princeton University
[3]Department of Statistics, Stanford University

The ability to switch between different tasks is a critical component of adaptive cognitive functioning, but a mechanistic understanding of this capacity has remained elusive. Longstanding debates over whether task switching requires active preparation remain hotly contested, in large part due to the difficulty of inferring task preparation from behavior alone. We make progress on this debate by quantifying neural task representations through high-dimensional linear dynamical systems fit to human electroencephalographic recordings. We find that these dynamical systems are highly predictive of macroscopic neural activity, and reveal neural signatures of active preparation that are shared with task-optimized neural networks. These findings help inform a classic debate about how we control our cognition, and offer a promising new paradigm for neuroimaging analysis.

task switching | electroencephalography | state space model | recurrent neural network

Correspondence: ***hr0283 @princeton.edu***

We have a remarkable capacity to flexibly change how we perform tasks (1–4), such as switching between speaking different languages. This flexibility reflects a core component of goal-directed cognition (5) and is a strong indicator of both cognitive development (6, 7) and mental health (8, 9). Despite the centrality of cognitive flexibility in our mental life, we still lack a mechanistic understanding of how people can rapidly configure task processing to achieve their moment-to-moment goals.

Separate literatures in psychology and neuroscience have explored how people flexibly prepare for upcoming decisions or actions. In cognitive psychology, there is a rich history of research exploring how we switch between tasks, often probed using the performance costs that accompany rapidly switching stimulus-response mappings (switching of 'task sets'; (2)). A longstanding debate in this field hinges on whether these switch costs, and the underlying processes they reflect, are due to 'passive inertia' or 'active reconfiguration' (3). The inertia hypothesis argues that switch costs largely reflect interference from the previous task set, which passively decays over time (10). The reconfiguration hypothesis argues instead that switch costs reflect active preparation for the upcoming task, which is dependent on cognitive control (1, 11). Behavioral experiments have attempted to adjudicate between these hypotheses (12–14), but such studies are limited in their ability to characterize how latent preparatory representations evolve over time.

In parallel, a major ongoing focus in the neuroscience of *motor* control has been how humans and other animals prepare upcoming actions. A major recent advance in this field has been to use dynamical systems theory to explain how populations of motor neurons reach an optimal preparatory brain state (15–18). This framework offers a promising solution for cognitive control, as it aligns with recent cognitive theories of how people dynamically transition between *task* states (19–21). However, these dynamical formalisms of task switching have yet to be exploited to explain how people reach task-appropriate brain states.

The current project addresses both of the closely-related questions raised above. First, how can we measure preparatory brain states for cognitive tasks, to adjudicate between task-switching hypotheses? We test the utility of the dynamical systems perspective for answering this question, using a novel application of state space modeling to quantify the large-scale dynamics of human brain activity. Second, do neural dynamics support the inertia or active reconfiguration hypotheses? We use this new modeling approach to directly test competing hypotheses about whether task switching is better described by passive decay or by active reconfiguration. Formalizing these two hypotheses using task-optimized recurrent neural networks (RNNs), we find that common dynamical motifs across biological and artificial neural systems support the active reconfiguration of latent task sets.

Traditional multivariate analyses reveal dynamic task encoding

We analyzed scalp electroencephalography (EEG) recordings made in a recent experiment (22) in which thirty participants performed a cued task-switching paradigm (Figure 1A; Method 1 and 2). On each trial, participants were cued whether to perform a 'shape' or 'color' task, with two potential cues per task. After a delay period, participants saw a colored shape, and responded based on the cued dimension (shape or color) with a button press. The original article reported robust EEG encoding of the task identity during the pre-trial period, which was interpreted as reflecting a preparatory brain state that facilitates task-appropriate behavior (23–25).

We extended these traditional multivariate EEG analyses to assess whether task encoding differed between trials in which participants switched tasks compared to when they repeated tasks. The inertia hypothesis predicts poorer task encoding on switch trials than repeat trials, due to interference with the previous task set. By contrast, the reconfiguration hypothesis predicts *stronger* task encoding on switch trials due to the deployment of cognitive control (25, 29). Using a recently-developed multivariate encoding analysis (encoding geome-

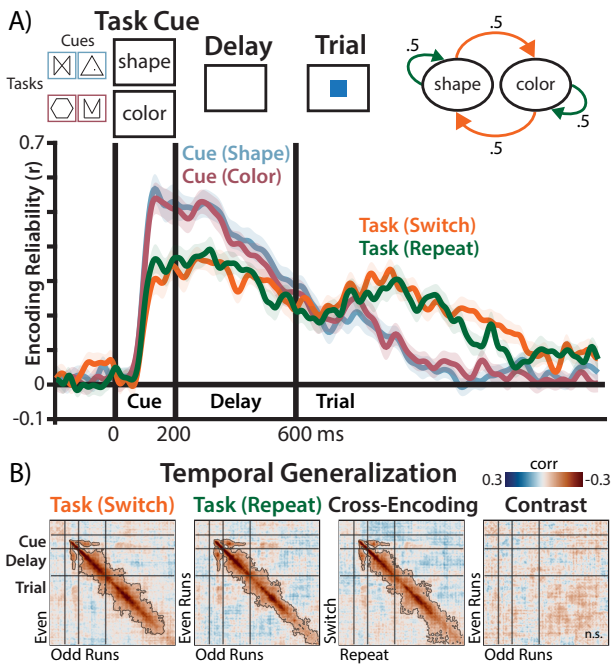

Fig. 1. Sensor-level encoding. **A)** Top, participants saw a task cue (two symbols mapped each task), waited through a delay period, then performed the task. Bottom: encoding reliability of cues (red and blue) and tasks (green and orange) during the post-cue period. Traces are smoothed for visualization. Error bars reflect within-subject SEM (26). **B)** Temporal generalization of task patterns, correlating task patterns between different timepoints (and across data splits). For switch and repeat panels, the diagonal is the same as the traces in (A). 'Cross-Encoding' depicts the similarity of task patterns between switch and repeat trials. 'Contrast' depicts the difference in encoding strength between switch and repeat trials. Contour indicates $p < .05$, corrected using threshold-free cluster enhancement (TFCE, (27, 28)).

try analysis; EGA; (30)), we found that task identity was robustly encoded throughout the preparatory epoch and into the trial period (Figure 1A). However, sensor-level EGA demonstrated similar task-encoding strength for switch and repeat trials, inconsistent with the predictions from both theories.

To better understand whether there is potential dynamical structure missed by time-locked encoding analyses, we next explored how task-encoding changed across the epoch. We tested for the presence of encoding dynamics using temporal generalization analysis (31), measuring the cross-validated similarity between task patterns at different temporal lags (Method 4). These analyses revealed poor generalization of task representations beyond a few hundred milliseconds (Figure 1B), strongly suggesting that task encoding dynamically evolved across the epoch. Again inconsistent with both tasks-switching hypotheses, task encoding did not significantly differ between switch and repeat trials at any temporal lag.

In summary, EGA was unable to discriminate between the inertia and reconfiguration theories, as both theories predict differences in task encoding between switch and repeat trials. However, temporal generalization analyses revealed that task encoding is highly dynamic. This suggests that we might be able to better discern potential differences between switch and repeat trials by examining in greater detail the temporal structure of the underlying task representations.

Inferring neural state space dynamics

To characterize the rich dynamics of task encoding suggested by EGA, we re-analyzed this EEG dataset using (linear) state space analysis (SSA). In contrast to EGA, this approach explicitly accounts for the temporal structure of task representations by modeling the EEG recordings as a linear dynamical system over a set of latent factors.

SSA has long been used in computational neuroscience to model low-dimensional dynamics in multi-unit neural recordings (32–35). SSA provides a parsimonious alternative to EEG analyses using nonlinear (36, 37) or discrete-state (38, 39) dynamical systems. Despite the nonlinearity of neural activation functions, recent work has suggested that linear models are effective models of neural dynamics (40–42). Critical for our purposes, linear models provide access to a suite of tools from dynamical systems theory and control theory for understanding the effects of recurrent connectivity and inputs (43), and help bridge neural and cognitive theories relying on similar dynamical formalisms.

SSA infers a set of latent factors (state vector $\mathbf{x}$; Figure 2A; Method 5) that evolve according to a standard autoregressive process. Factor dynamics are driven by three components: the recurrence matrix (A), known inputs (input vector $\mathbf{u}$, projected through the input matrix B), and Gaussian process noise ($\mathbf{w}_t \sim \mathcal{N}(0, W)$). The dynamics on each trial evolve according to:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_{t-1} + \mathbf{w}_t \quad \textbf{(1)}$$

To test our core hypothesis about the nature of task encoding on switch versus repeat trials, our model received the current task identity as an input (separately for switch and repeat trials; see Method 6 for full predictor list). We accounted for time-varying neural processes below the resolution of EEG by expanding our trial-wise regressors ($\mathbf{r}_n$) with a smooth temporal basis function ($\mathbf{u}_{t,n} = \mathbf{\Phi}_t \otimes \mathbf{r}_n$; cubic B-spline basis; Figure 2A; Method 6). To model the carry-over from the previous trial, the initial condition depended on the previous task (see Method 6 for predictor list) and initialization noise ($\mathbf{x}_1 = B_0\mathbf{u}_0 + \mathbf{w}_1$).

Factors were assumed to have a linear relationship with the EEG recordings ($\mathbf{y}$), with factors 'read-out' by the observation matrix (C) and then corrupted by Gaussian observation noise ($\mathbf{v}_t \sim \mathcal{N}(0, V)$)

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t \quad \textbf{(2)}$$

We reduced the dimensionality of the EEG observations during preprocessing using principal components analysis (PCA), retaining 14 – 27 components (99% of the variance). We modeled neural task preparation across the cue period, delay period, and the first 200ms of the trial (before any response) across all trials (i.e., without averaging). We estimated the maximum *a posteriori* parameters of this model through a standard expectation-maximization fitting

procedure (44, 45), implemented in a new toolkit called `StateSpaceAnalysis`.

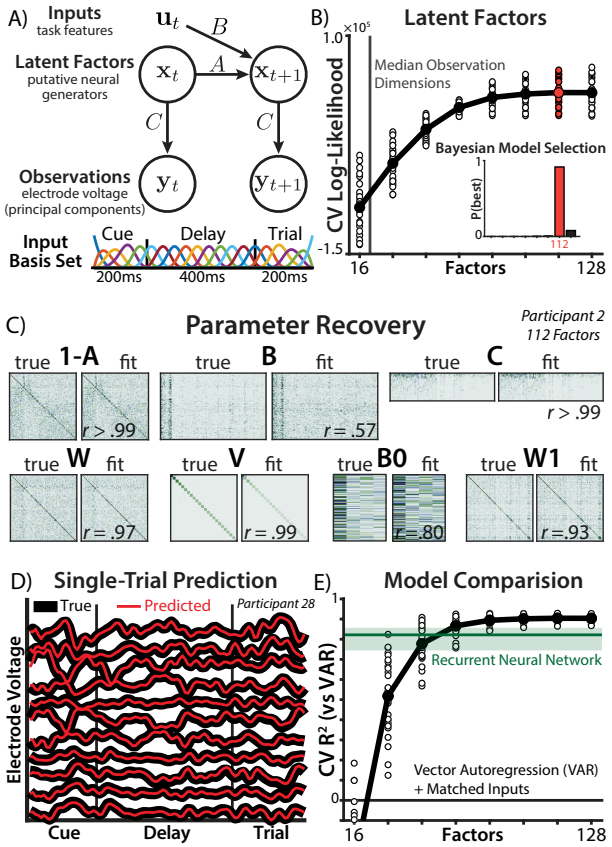

Fig. 2. State Space Analysis. **A)** Top : Schematic of the latent state space model. Bottom: temporal basis set spanning the epoch. **B)** The log-likelihood from a held-out test set (y-axis), across different number of factors (x-axis). Dots indicate single subjects; log-likelihood is mean-centered within-participant to show relative differences. Inset: Bayesian model selection. **C)** Parameter recovery on a synthetic dataset generated from participant 2's parameters. **D)** Predictive accuracy in one example trial from the test set across a subset of the EEG electrodes. Thick black lines indicate EEG voltage, red lines indicate next-timestep predictions from a Kalman filter (112 factor model). **E)** Cross-validated coefficient of determination for SSA models across factor size, relative to autoregressive encoding models fit directly to the observations. Horizontal green line indicates the fit of parameter-matched RNNs; shaded area indicates the 95% interval of the RNN fit across participants.

EEG recordings are well-explained by a high-dimensional linear dynamical system

We first assessed the dimensionality of our latent space, choosing the number of factors based on cross-validated predictive accuracy. In systems neuroscience, latent variable models have typically found fewer factors than observation dimensions (46, 47). Surprisingly, we found that SSA models with more factors than the original number of electrodes provided the best cross-validated log-likelihood (best model: 112 factors with 90% protected exceedance probability; Figure 2B; (48)). Speculatively, this 'lifting' of the dimensionality may allow for a good linear approximation to an underlying nonlinear system (49).

To validate the reliability of this over-parameterized model, we measure SSA's performance on a synthetic dataset. SSA was highly accurate at recovering the ground truth parameters (Figure 2C; after factor alignment, see Method 6). This strong recovery supports not only the accuracy of SSA as a procedure, but also our ability to differentiate effects across our parameters.

Consistent with this good recovery, our best-fitting model had highly accurate next-timestep prediction, evident even in single trials from the test set (Figure 2D). This predictive accuracy was maintained when forecasting 100s of milliseconds into the future (median $t+8$ms accuracy: $R^2_{CV} = 98\%$; median $t+200$ms accuracy: $R^2_{CV} = 40\%$; Figure S1). Model comparison revealed that SSA performed better than both simpler autoregressive encoding models and non-linear recurrent neural network models (Figure 2E; Method 7), suggesting that SSA strikes a good balance between linear models' interpretability and non-linear models' expressivity.

Latent neural dynamics reveal stable task control

Having validated the predictive power of SSA, we turned to characterizing the dynamics of task preparation. We first assessed whether participants' latent neural trajectories support an important untested prediction from dynamic theories of task switching: convergence towards a stable, task-specific state that facilitates good performance (19, 20).

To assess the evolution of such task states, we used a procedure known as 'additive state decomposition', which factorizes a linear dynamical systems into a set of subsystems for each input ((50); Method 8). We simulated the subsystem for time-varying task inputs ($\mathbf{x}_t^{\text{task}} = A\mathbf{x}_{t-1}^{\text{task}} + B^{\text{task}}\mathbf{\Phi}_{t-1}$), with 'previous task' inputs setting the initial conditions.

We visualized the simulated task trajectories using singular value decomposition (SVD) to project participants' data into a common low-dimensional space (Method 8). We found that the same task appears to have similar trajectories across switch and repeat conditions, with states for the same task becoming more similar over time (Figure 3A).

To quantify how task states converged over time, we measured the Euclidean distances between states across switch and repeat conditions (Method 8). Whereas brain states for the same task became more similar over time (Figure 3B, black), states for different tasks diverged during the cue period and then maintained separation (Figure 3B, pink). Both of these dynamics support convergence to a task-specific brain state.

Task-switching theories predict that task-specific states should be stable (e.g., reflecting an attractor state (51)). We quantified the stability of task representations using temporal generalization analysis (compare to Figure 1B), finding that generalization appeared to increase over epoch (Figure 3C; off-diagonal similarity). Testing whether off-diagonal similarity increased over time, we found significant generalization increases for temporal lags up to 248 ms (see Figure S2).

Finally, we investigated whether these stable task states facilitate upcoming performance. Consistent with preparation,

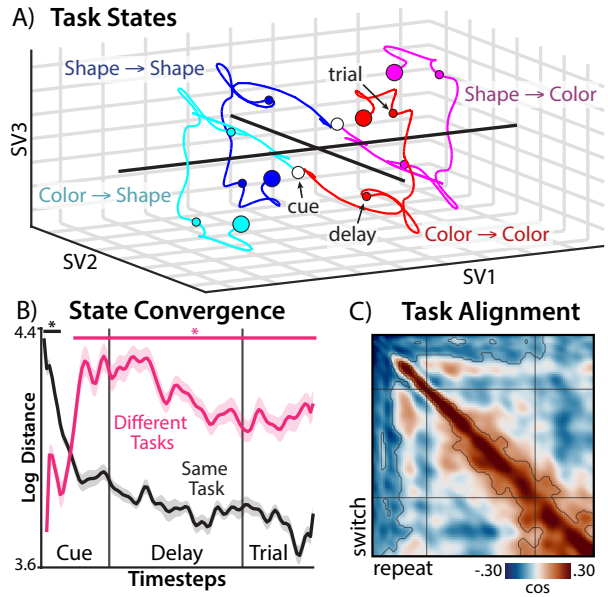

Fig. 3. Latent task dynamics. **A)** Simulated task states for each task, split by switch and repeat conditions. Since task is contrast-coded, the two tasks within each switch condition are symmetrical. States are projected to 3D using SVD on participant-concatenated trajectories (see Method 8). **B)** Task state distances across switch and repeat conditions. Black line indicates the distance between the same task across switch and repeat simulations. Pink line indicates the distance between different tasks across switch and repeat simulations. Asterisks reflect TFCE-corrected significant differences. **C)** Temporal generalization of task states across switch and repeat trials. Vertical and horizontal lines indicate the the boundaries between the cue, delay, and trial periods (Compare to Figure 1B). Contour reflect p < .05 (TFCE-corrected).

we found that participants had faster reaction times on trials for which the EEG signal was more similar to the task state ($d = -0.53$, $p = .0068$; Figure S3; Method 8). Together, these findings support key tenets of dynamical theories of task switching: the brain transitions between stable, task-specific states that promote good performance.

Task dynamics support active reconfiguration

We found that neural task trajectories supported the core tenets of dynamical task switching theories. We next examined whether these neural dynamics can help discriminate between the inertia and reconfiguration theories of task switching, which we could not distinguish using traditional sensor-level encoding analyses (see Figure 1).

These task switching hypotheses critically differ in how task states evolve during switch and repeat trials. The inertia hypothesis predicts weaker task encoding on switch trials than repeat trials, due to passively decaying interference from the previous trial. The reconfiguration hypothesis predicts stronger encoding on switch trials, due to active changes to the state. We operationalized each hypothesis through estimates of the strength and dynamics of task encoding.

To measure the strength of task encoding, we drew on a metric from control theory termed the 'average controllability Gramian' (52–54). The controllability Gramian defines the influences of an input by how much covariance it causes in the system ($C_\infty = AC_\infty A^\top + BB^\top$; see Method 8), and is commonly summarized with the matrix trace (52). We adapted this metric to capture the strength of time-varying task inputs over the epoch, which we call the 'task Gramian' (Method 8):

$$G_t = AG_{t-1}A^\top + (B^{\text{task}}\mathbf{\Phi}_{t-1})(B^{\text{task}}\mathbf{\Phi}_{t-1})^\top \quad \textbf{(3)}$$

To adjudicate between task-switching hypotheses, we tested whether the average task Gramian was stronger on switch trials or repeat trials. In contrast to our traditional analyses, here we found that there was a stronger task Gramian when participants switched tasks (Figure 4A; summed over timesteps: $d = 0.83$, $p = 8.6 \times 10^{-5}$), supporting the reconfiguration hypothesis.

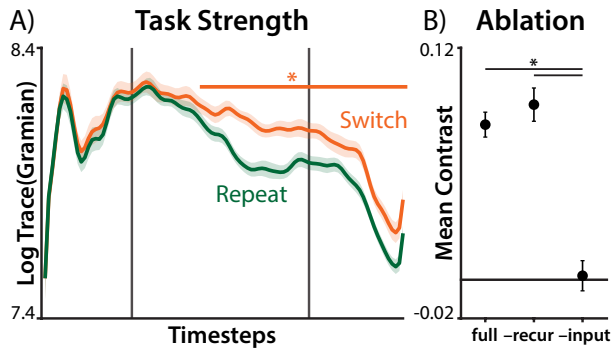

Fig. 4. Latent task dynamics. **A)** The log trace of the task Gramian across the epoch, separately for switch (orange) and repeat (green) conditions. Error bars indicates within-participant SEM, asterisk indicates significant differences between switch and repeat conditions (TFCE-corrected). **B)** The time-averaged contrast between switch and repeat Gramians, under different model ablations. The full model had no reduction, reflecting the time-averaged effect in (A). The '–recur' model has the fitted recurrence and initial conditions matrices set to zero. The '–inputs' model has normalized inputs at each timestep (2-norm). Error bars reflect within-participant SEM, asterisk indicates $p < .05$

Task switching theories make predictions not only about the strength of task encoding, but also about the dynamics of task encoding. We operationalized passive decay in the inertia hypothesis as how the initial conditions (i.e., previous task input) spread through the system via the recurrence matrix (Method 8). However, we found that setting the fitted initial conditions and recurrence to zero did not reduce the difference in task strength between switch conditions (Figure 4B).

We operationalized active switching in the reconfiguration hypothesis as the time-varying magnitude of task inputs. Here, we found that normalizing the fitted task inputs at each timestep was sufficient to eliminate task strength differences between switch conditions (Figure 4B), again supporting the reconfiguration account.

These task encoding analyses provided consistent support for dynamical theories of task switching in general, and the reconfiguration hypothesis in particular. We found that time-varying inputs move the brain into stable task states, especially when switching between tasks. We hypothesized that these time-varying task inputs reflect an underlying control cognitive process (19, 20). To facilitate a stronger interpretation of this potential role for control, we formalized the in-

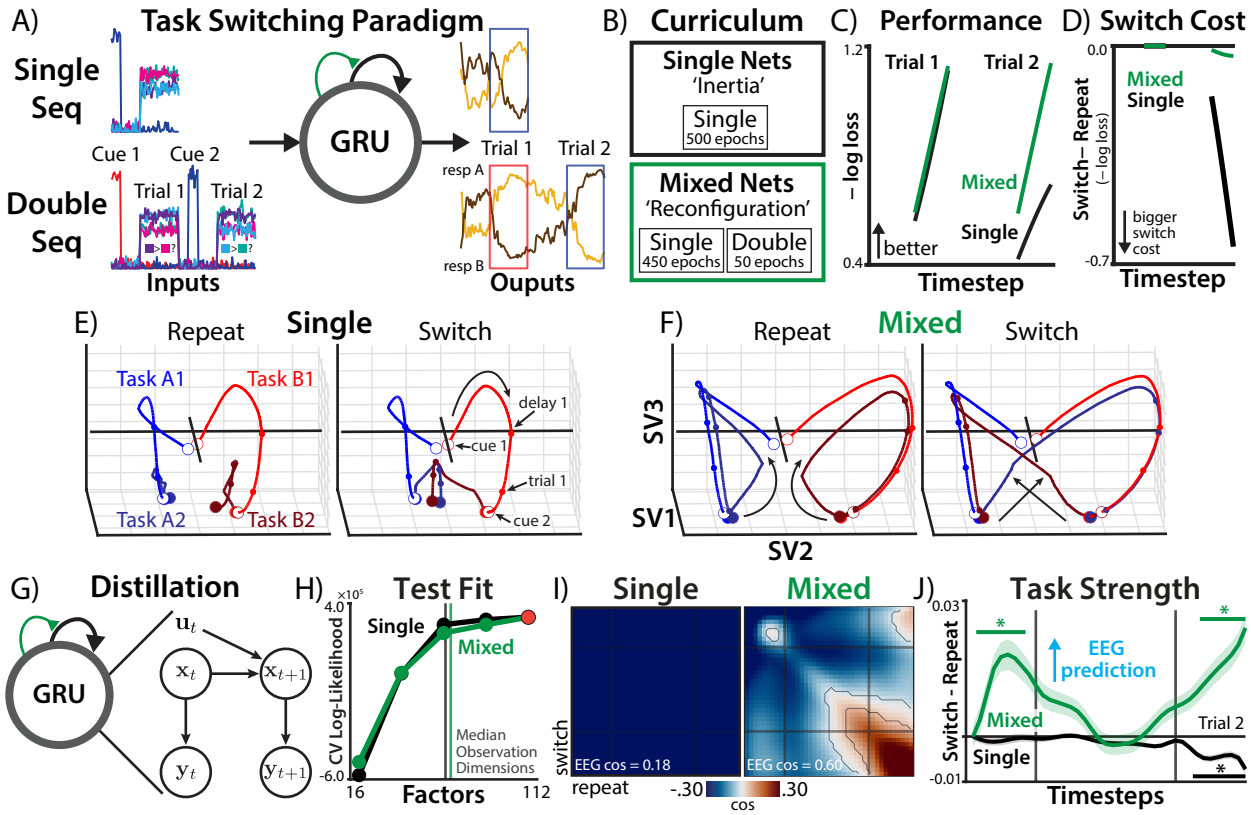

Fig. 5. Task dynamics in neural networks. **A)** examples of single sequences (top) and double sequences (bottom), which differed in whether the required engaging in more than one task in a sequence. **B)** Single networks were only trained on single sequences, whereas mixed networks had 10% training on double sequences. **C)** network loss over the course of three-trial test sequences. Error bars are smaller than line width. **D)** Difference in within-sequence loss for switch and repeat trials. **E-F)** Hidden unit activity during two-trial sequences projected to three dimensions with SVD. Note that the SVD was performed across single (D) and mixed (E) networks. White dots are the initial timestep of each trial, colored dots are the final timestep of each trial. **G)** State space analysis of GRU hidden unit trajectories. **H)** Cross-validated log-likelihood across different numbers of SSA latent dimensions. **I)** Temporal generalization of task states across switch and repeat trials. Contour indicates $p < .05$ (TFCE-corrected). Inset statistics are from pixel-wise correlating mean similarity matrices between down-sampled human EEG and each network. **J)** Contrast between task strength between switch and repeat trials. Mixed models show the same contrast direction as human EEG, unlike single models. Asterisks reflect differences from zero at $p < .05$, corrected using TFCE.

ertia and reconfiguration hypotheses using gated RNNs, and then compared networks' task encoding dynamics to those revealed by the EEG analyses reported above.

Task reconfiguration in recurrent neural networks

We formalized the inertia and reconfiguration hypotheses using RNNs, expressive dynamical systems that can learn to perform basic cognitive tasks (55). Previous neural network models of task switching have implemented inertia and reconfiguration theories through hand-crafted structural mechanisms (19, 56–58), and have not adjudicated between these theories on the basis of their neural predictions. We instead implemented these theories through different training curricula (20, 59), comparing networks' optimized task-encoding dynamics to human EEG.

We trained networks on a context-dependent decision-making task (60–62), which closely mirrored the EEG task (Method 9). There were two different types of sequences. During 'single sequences', networks received a task cue, waited through a delay period, and then responded based on the task-relevant stimulus features (Figure 5A, upper). Whereas single sequences only had one trial per sequence, during 'double sequences', networks performed two consecutive trials per sequence (Figure 5A, lower), with equal probability of switching and repeating tasks.

We modeled the two task switching hypotheses through networks' experience with single and double sequences (Figure 5B; Method 9). The inertia hypothesis was captured by 'single networks', which had 100% of their training on single sequences. We hypothesized that these networks would show passive decay in the absence of learning how to switch. The reconfiguration hypothesis was captured by 'mixed networks', which had 90% of their training on single sequences and the remaining 10% on double sequences. We hypothesized that this limited switching experience would first encourage reconfiguration strategies, before changes to well-learned task representations.

Switching practice teaches flexible reconfiguration strategies

We first explored whether single and mixed networks offer face-valid reproductions of the inertia and reconfiguration theories, training 512 RNNs with gated recurrent units

(GRUs) under each curriculum (Method 9). Evaluating the networks performance on double sequences, we plotted the average loss at each timestep. We found that single and mixed networks had similar performance on the first trial, suggesting similar capacity to perform the tasks (Figure 5C). However, mixed networks showed much stronger performance on the second trial, which they had been trained on. Mixed networks' advantage on second trials was in part due to better switching, as mixed networks had much lower switch costs for their loss than did single networks (Figure 5D), consistent with learning an adaptive reconfiguration strategy.

We assessed whether mixed networks learned generalizable switching strategies, as one would expect from flexible cognitive control. First, we tested how switching generalized to new sequences. Testing fitted networks on novel 'triple sequences', we found that the mixed networks still performed better than single networks on untrained third trials (Figure S4). Second, we tested how switching generalized to new tasks. We trained both networks on four tasks, and then trained mixed networks to switch between only two of the tasks (Method 9). After training, we found that mixed networks had better switch-trial performance than single networks on the two tasks that they weren't trained to switch between, consistent with learning a generalizable reconfiguration strategy (Figure S5).

Finally, we tested the importance of gating mechanisms for switching, given their hypothesized analogy to cognitive control (63, 64). We found that ablating the GRU 'update' gate substantially impaired these networks' ability to learn from double sequences (Figure S6), consistent with a role for control in reconfiguration.

Quantifying RNN reconfiguration with linear dynamical systems

We found that task-optimized RNNs adopt flexible switching strategies, allowing us to compare models that learn to perform tasks (cf. inertia) to models that learn to switch tasks (cf. reconfiguration). A major benefit of RNN modeling is that we have full access to the underlying representations, letting us compare networks' optimized dynamics to the latent trajectories in human EEG.

We first visualized RNNs' hidden-unit activations using SVD (compare to Figure 3A), projecting a single network and mixed network into a shared low-dimensional space (Method 9). Whereas single and mixed networks exhibited similar first-trial dynamics, their second-trial dynamics sharply diverged. Single networks appeared to move directly between the two trial states, ending up in the middle (Figure 5E). By contrast, mixed networks exhibited much more robust cued-evoked dynamics (Figure 5F).

To compare the task switching dynamics in each of these networks to human EEG, we used SSA to fit linear dynamics systems to RNNs' hidden unit activations during preparation for the second trial (Figure 5G; Method 10). Similar to our EEG analyses, we found that high-dimensional latent states improved cross-validated fit (Figure 5H) and provided accurate predictions (one-step $R^2_{CV} = .90$; Figure S7).

We first used temporal generalization analyses to compare the timecouse of task representations across RNNs and EEG (Figure 5I, compare to Figure 2C). In single networks, task states across switch and repeat conditions were negatively aligned throughout the epoch. While this is inconsistent with the EEG results, this plausibly reflects the inertia hypothesis's core prediction of cross-task interference. By contrast, mixed networks demonstrated a profile that was much more similar to human EEG, with growing stability through the epoch. Compare the temporal generalization profiles across networks and EEG, we found that mixed networks were more similar to EEG than were single networks (cosine between mean similarity matrices: $\cos$(Mixed, EEG) = .60, $\cos$(Single, EEG) = .18; difference $p < .001$; Method 10), supporting the reconfiguration account.

Finally, we tested whether RNNs replicated our EEG signature of stronger task encoding on switch trials than repeat trials. We found that single and mixed networks exhibited opposite patterns (Figure 5J). Single networks, unlike humans, had weaker task strength on switch trials than repeat trials (summed over timesteps: $d = -0.091$, $p = .040$). Mixed networks, consistent with human EEG, had stronger task encoding on switch trials than repeat trials ($d = 0.15$, $p = .0010$). This concurrence between switching-optimized RNNs and human EEG provides further evidence supporting the reconfiguration account of task switching.

Discussion

This experiment demonstrates the power of a dynamical systems approach for understanding cognitive flexibility in particular, and the trajectories of neural representations more generally. Linear dynamical systems fit to biological and artificial neural networks provided new insight into the active reconfiguration of neural states, contributing to a decades-old debate in cognitive control.

One prominent feature in both EEG and neural networks was robust configuration for both switch and repeat trials (e.g., Figure 3A and Figure 5F), suggestive of a larger role for active configuration than has previously been assumed (2). One caveat to this proposal is that the present study had longer inter-trial intervals than are typically used. While debates over task switching theories have included paradigms with long delays (3, 65), closely-spaced tasks may reveal addition pressures on preparatory neural activity.

To better characterize the underlying task control process, future work should model the origins of time-varying task inputs. Within a dynamical systems framework, a natural candidate is optimal control theory (52, 53). For cognitive neuroscience more broadly, there is a high demand for better spatial localization of the neural generators for EEG signals. State space models may provide powerful methods for localizing latent dynamics (66, 67), offering a better view into the distributed neural systems enabling our cognitive control (68).

ACKNOWLEDGEMENTS

We are grateful to Sam Hall-McMaster for making this dataset public and providing additional support, Christopher Langdon for sharing PyTorch code for the context-dependent decision-making task, and Caroline Jahn for feedback on a draft of the manuscript. This paper is dedicated to Mark Stokes.

HR was supported by the C.V. Starr Postdoctoral Fellowship. AJ was supported by the Google PhD fellowship. JWP was supported by grants from the Simons Collaboration on the Global Brain (SCGB AWD543027) and the NIH BRAIN initiative (9R01DA056404-04,U19NS104648,U19NS123716,9R01DA056404-04). JDC was supported by a Vannevar Bush Faculty Fellowship administered by the Office of Naval Research.

Bibliography

1. Stephen Monsell. Task switching. *Trends Cogn. Sci.*, 7(3):134–140, March 2003.
2. Andrea Kiesel, Marco Steinhauser, Mike Wendt, Michael Falkenstein, Kerstin Jost, Andrea M Philipp, and Iring Koch. Control and interference in task switching—a review. *Psychol. Bull.*, 136(5):849–874, September 2010.
3. André Vandierendonck, Baptist Liefooghe, and Frederick Verbruggen. Task switching: interplay of reconfiguration and interference control. *Psychol. Bull.*, 136(4):601–626, July 2010.
4. Sebastian Musslick and Jonathan D Cohen. Rationalizing constraints on the capacity for cognitive control. *Trends Cogn. Sci.*, 0(0), July 2021.
5. Naomi P Friedman and Akira Miyake. Unity and diversity of executive functions: Individual differences as a window on cognitive structure. *Cortex*, 86:186–204, January 2017.
6. Mark Steyvers, Guy E Hawkins, Frini Karayanidis, and Scott D Brown. A large-scale analysis of task switching practice effects across the lifespan. *Proc. Natl. Acad. Sci. U. S. A.*, August 2019.
7. Nicholas J Cepeda, Arthur F Kramer, and Jessica C M Gonzalez de Sather. Changes in executive control across the life span: Examination of task-switching performance. *Dev. Psychol.*, 37(5):715–730, 2001.
8. Mark J Millan, Yves Agid, Martin Brüne, Edward T Bullmore, Cameron S Carter, Nicola S Clayton, Richard Connor, Sabrina Davis, Bill Deakin, Robert J DeRubeis, Bruno Dubois, Mark A Geyer, Guy M Goodwin, Philip Gorwood, Thérèse M Jay, Marian Joëls, Isabelle M Mansuy, Andreas Meyer-Lindenberg, Declan Murphy, Edmund Rolls, Bernd Saletu, Michael Spedding, John Sweeney, Miles Whittington, and Larry J Young. Cognitive dysfunction in psychiatric disorders: characteristics, causes and the quest for improved therapy. *Nat. Rev. Drug Discov.*, 11(2):141–168, February 2012.
9. Hannah R Snyder. Major depressive disorder is associated with broad impairments on neuropsychological measures of executive function: a meta-analysis and review. *Psychol. Bull.*, 139(1):81–132, January 2013.
10. A Allport, E A Styles, and S L Hsieh. Shifting intentional set - exploring the dynamic control of tasks. In *ATTENTION AND PERFORMANCE XV: CONSCIOUS AND NONCONSCIOUS INFORMATION PROCESSING*, pages 421–452. MIT Press, London, England, 1994.
11. Robert D Rogers and Stephen Monsell. Costs of a predictible switch between simple cognitive tasks. *J. Exp. Psychol. Gen.*, 124(2):207, 1995.
12. Nachshon Meiran. Reconfiguration of processing mode prior to task performance. *J. Exp. Psychol. Learn. Mem. Cogn.*, 22(6):1423–1442, November 1996.
13. Nick Yeung and Stephen Monsell. Switching between tasks of unequal familiarity: the role of stimulus-attribute and response-set selection. *J. Exp. Psychol. Hum. Percept. Perform.*, 29(2):455–469, April 2003.
14. Florian Schmitz and Andreas Voss. Decomposing task-switching costs with the diffusion model. *J. Exp. Psychol. Hum. Percept. Perform.*, 38(1):222–250, February 2012.
15. Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annu. Rev. Neurosci.*, 36:337–359, July 2013.
16. Jonathan A Michaels, Benjamin Dann, and Hansjörg Scherberger. Neural population dynamics during reaching are better explained by a dynamical system than representational tuning. *PLoS Comput. Biol.*, 12(11):e1005175, November 2016.
17. Marine Schimel, Ta-Chu Kao, and Guillaume Hennequin. When and why does motor preparation arise in recurrent neural network models of motor control? August 2023.
18. Mark M Churchland and Krishna V Shenoy. Preparatory activity and the expansive null-space. *Nat. Rev. Neurosci.*, 25(4):213–236, April 2024.
19. Salva Ardid and Xiao-Jing Wang. A tweaking principle for executive control: neuronal circuit mechanism for rule-based task switching and conflict resolution. *J. Neurosci.*, 33(50): 19504–19517, December 2013.
20. Sebastian Musslick, Seong Jun Jang, Michael Shvartsman, Amitai Shenhav, and Jonathan D Cohen. Constraints associated with cognitive control and the stability-flexibility dilemma. In *CogSci*. shenhavlab.org, 2018.
21. Paul I Jaffe, Russell A Poldrack, Robert J Schafer, and Patrick G Bissett. Modelling human behaviour in cognitive tasks with latent dynamical systems. *Nature Human Behaviour*, pages 1–15, January 2023.
22. Sam Hall-McMaster, Paul S Muhle-Karbe, Nicholas E Myers, and Mark G Stokes. Reward boosts neural coding of task rules to optimize cognitive flexibility. *J. Neurosci.*, 39(43):8549–8561, October 2019.
23. J D Cohen, K Dunbar, and J L McClelland. On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychol. Rev.*, 97(3):332–361, July 1990.
24. J D Wallis, K C Anderson, and E K Miller. Single neurons in prefrontal cortex encode abstract rules. *Nature*, 411(6840):953–956, June 2001.
25. Alexandra Woolgar, Adam Hampshire, Russell Thompson, and John Duncan. Adaptive coding of task-relevant information in human frontoparietal cortex. *J. Neurosci.*, 31(41): 14592–14599, October 2011.
26. Richard D Morey. Confidence intervals from normalized data: A correction to cousineau. In *Tutorials in Quantitative Methods for Psychology , 4*. Citeseer, 2008.
27. Stephen M Smith and Thomas E Nichols. Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. *Neuroimage*, 44(1):83–98, January 2009.
28. Armand Mensen and Ramin Khatami. Advanced EEG analysis using threshold-free cluster-enhancement and non-parametric statistics. *Neuroimage*, 67:111–118, February 2013.
29. J A Etzel, M W Cole, J M Zacks, K N Kay, and T S Braver. Reward motivation enhances task coding in frontoparietal cortex. *Cereb. Cortex*, 26(4):1647–1659, April 2016.
30. Harrison Ritz and Amitai Shenhav. Orthogonal neural encoding of targets and distractors supports multivariate cognitive control. *Nature Human Behaviour*, pages 1–17, March 2024.
31. J-R King and S Dehaene. Characterizing the dynamics of mental representations: the temporal generalization method. *Trends Cogn. Sci.*, 18(4):203–210, April 2014.
32. E N Brown, L M Frank, D Tang, M C Quirk, and M A Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, 18(18):7411–7425, September 1998.
33. Anne C Smith and Emery N Brown. Estimating a state-space model from point process observations. *Neural Comput.*, 15(5):965–991, May 2003.
34. Jakob H Macke, Lars Büsing, John P Cunningham, Byron M Yu Ece, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In *Advances in neural information processing systems*, 2011.
35. Scott Linderman, Annika Nichols, David Blei, Manuel Zimmer, and Liam Paninski. Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in C. elegans. *bioRxiv*, page 621540, April 2019.
36. Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. In *International Conference on Machine Learning*, pages 13017–13049. PMLR, July 2023.
37. Matthijs Pals, A Erdem Sağtekin, Felix Pei, Manuel Gloeckler, and Jakob H Macke. Inferring stochastic low-rank recurrent neural networks from neural data. *arXiv [cs.LG]*, June 2024.
38. D Lehmann, H Ozaki, and I Pal. EEG alpha map series: brain micro-states by space-oriented adaptive segmentation. *Electroencephalogr. Clin. Neurophysiol.*, 67(3):271–288, September 1987.
39. Chetan Gohil, Oliver Kohl, Rukuang Huang, Mats W J van Es, Oiwi Parker Jones, Laurence T Hunt, Andrew J Quinn, and Mark W Woolrich. Dynamic network analysis of electrophysiological task data. *Imaging Neuroscience*, 2:1–19, July 2024.
40. Erfan Nozari, Maxwell A Bertolero, Jennifer Stiso, Lorenzo Caciagli, Eli J Cornblath, Xiaosong He, Arun S Mahadevan, George J Pappas, and Dani S Bassett. Macroscopic resting-state brain dynamics are best described by linear models. *Nat. Biomed. Eng.*, pages 1–17, December 2023.
41. Joana Soldado-Magraner, Valerio Mante, and Maneesh Sahani. Inferring context-dependent computations through linear approximations of prefrontal cortex dynamics. *bioRxiv*, page 2023.02.06.527389, February 2023.
42. Yizi Zhang, Hanrui Lyu, Cole Hurwitz, Shuqi Wang, Charles Lincoln Findling, Felix Hubert, Alexandre Pouget, International Brain Laboratory, Erdem Varol, and Liam Paninski. Exploiting correlations across trials and behavioral sessions to improve neural decoding. *bioRxivorg*, page 2024.09.14.613047, September 2024.
43. Robert L Williams and Douglas A Lawrence. *Linear state-space control systems*. John Wiley & Sons, Nashville, TN, January 2007.
44. Zoubin Ghahramani, Geo Rey, and E Hinton. Parameter estimation for linear dynamical systems. Techinical Report, 1996.
45. Kevin P Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, London, England, 2023.
46. Mark M Churchland, Byron M Yu, Maneesh Sahani, and Krishna V Shenoy. Techniques for extracting single-trial activity patterns from large-scale neural recordings. *Curr. Opin. Neurobiol.*, 17(5):609–618, October 2007.
47. Camden J MacDowell, Brandy A Briones, Michael J Lenzi, Morgan L Gustison, and Timothy J Buschman. Differences in the expression of cortex-wide neural dynamics are related to behavioral phenotype. *Curr. Biol.*, February 2024.
48. Klaas Enno Stephan, Will D Penny, Jean Daunizeau, Rosalyn J Moran, and Karl J Friston. Bayesian model selection for group studies. *Neuroimage*, 46(4):1004–1017, July 2009.
49. Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv [math.DS]*, February 2021.
50. Olof Staffans. *Encyclopedia of mathematics and its applications: Well-posed linear systems series number 103*. Cambridge University Press, Cambridge, England, October 2009.
51. Jake P Stroud, John Duncan, and Máté Lengyel. The computational foundations of dynamic coding in working memory. *Trends Cogn. Sci.*, 0(0), April 2024.
52. E Tang and D S Bassett. Colloquium: Control of dynamics in brain networks. *Rev. Mod. Phys.*, 2018.
53. Ta-Chu Kao and Guillaume Hennequin. Neuroscience out of control: control-theoretic perspectives on neural circuit dynamics. *Curr. Opin. Neurobiol.*, 58:122–129, September 2019.
54. Clay B Holroyd. The controllosphere: The neural origin of cognitive effort. *Psychol. Rev.*, February 2024.
55. H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLoS Comput. Biol.*, 12(2):e1004792, February 2016.
56. Sam J Gilbert and Tim Shallice. Task switching: a PDP model. *Cogn. Psychol.*, 44(3): 297–337, May 2002.
57. Joshua W Brown, Jeremy R Reynolds, and Todd S Braver. A computational model of fractionated conflict-control mechanisms in task-switching. *Cogn. Psychol.*, 55(1):37–85, August 2007.
58. Seth A Herd, Randall C O'Reilly, Tom E Hazy, Christopher H Chatham, Angela M Brant, and Naomi P Friedman. A neural network model of individual differences in task switching abilities. *Neuropsychologia*, 62:375–389, September 2014.
59. J D Cohen, D Servan-Schreiber, and J L McClelland. A parallel distributed processing approach to automaticity. *Am. J. Psychol.*, 105(2):239–269, 1992.
60. Markus Siegel, Tobias H Donner, and Andreas K Engel. Spectral fingerprints of large-scale neuronal interactions. *Nat. Rev. Neurosci.*, 13(2):121–134, January 2012.
61. V Mante, D Sussillo, K V Shenoy, and W T Newsome. Context-dependent computation by

recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.
62. Christopher Langdon and Tatiana A Engel. Latent circuit inference from heterogeneous neural responses during cognitive tasks. *bioRxiv*, page 2022.01.23.477431, January 2022.
63. Todd S Braver and Jonathan D Cohen. Chapter 19 dopamine, cognitive control, and schizophrenia: the gating model. In *Progress in Brain Research*, volume 121 of *Progress in brain research*, pages 327–349. Elsevier, 1999.
64. Randall C O'Reilly and Michael J Frank. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Comput.*, 18(2):283–328, February 2006.
65. D A Allport, E A Styles, Shulan Hsieh, Carlo Umilta, and Morris Moscovitch. Attention and performance XV: Conscious and nonconscious information processing, 1994.
66. Andreas Galka, Okito Yamashita, Tohru Ozaki, Rolando Biscay, and Pedro Valdés-Sosa. A solution to the dynamical inverse problem of EEG generation using spatiotemporal kalman filtering. *Neuroimage*, 23(2):435–453, October 2004.
67. Elvira Pirondini, Behtash Babadi, Gabriel Obregon-Henao, Camilo Lamus, Wasim Q Malik, Matti S Hamalainen, and Patrick L Purdon. Computationally efficient algorithms for sparse, dynamic solutions to the EEG source localization problem. *IEEE Trans. Biomed. Eng.*, 65 (6):1359–1372, June 2018.
68. P S Goldman-Rakic. Topography of cognition: parallel distributed networks in primate association cortex. *Annu. Rev. Neurosci.*, 11:137–156, 1988.
69. Alexander Walther, Hamed Nili, Naveed Ejaz, Arjen Alink, Nikolaus Kriegeskorte, and Jörn Diedrichsen. Reliability of dissimilarity measures for multi-voxel pattern analysis. *Neuroimage*, 137:188–200, August 2016.
70. e Holmes, Elizabeth, j Ward, Eric, and Kellie Wills. MARSS: Multivariate autoregressive state-space models for analyzing time-series data. *R J.*, 4(1):11, 2012.
71. Gavin Smith, João de Freitas, Tony Robinson, and Mahesan Niranjan. Speech modelling using subspace and EM techniques. *Advances in Neural Information Processing Systems*, 12, 1999.
72. Iris R Stone, Yotam Sagiv, Il Memming Park, and Jonathan W Pillow. Spectral learning of bernoulli linear dynamical systems models. *arXiv [stat.ML]*, March 2023.
73. W E Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conference on Decision and Control*, pages 596–604 vol.2. IEEE, 1990.
74. Benedikt V Ehinger and Olaf Dimigen. Unfold: an integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis. *PeerJ*, 7(e7838):e7838, October 2019.
75. D R Cox and Nanny Wermuth. A comment on the coefficient of determination for binary responses. *Am. Stat.*, 46(1):1–4, February 1992.
76. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. *arXiv [cs.LG]*, December 2019.
77. G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.
78. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. November 2017.
79. R E Kalman. Lectures on controllability and observability. In *Controllability and Observability*, pages 1–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
80. Daniel B Ehrlich and John D Murray. Geometry of neural computation unifies working memory and planning. *Proc. Natl. Acad. Sci. U. S. A.*, 119(37):e2115610119, September 2022.
81. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv [cs.CL]*, June 2014.
82. Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.*, 2011(1):156869, January 2011.
83. Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods*, 134 (1):9–21, March 2004.
84. Fabio Crameri. Scientific colour maps, 2023.

Methods

1. EEG Sample

The EEG dataset analyzed here was originally published by Hall-McMaster and colleagues (Hall-McMaster et al., 2019), and was made open-access at https://osf.io/kuzye/. Thirty participants (18-35 years old; 19 females), with normal or corrected-to-normal vision and no history of neurological or psychological disorders, underwent an EEG recording session during task performance. This study was approved by the Central University Research Ethics Committee at the University of Oxford, with all participants providing informed consent. See (22) for full description.

2. EEG Task

Participants performed a cued task-switching experiment during the EEG recording. The major goal of the original study was investigating the role of incentives on task encoding. Our study focused on more traditional switching because this directly relates to our core theoretical questions about task set inertia and reconfiguration.

In the original experiment, the full trial structure involved a cue for high vs low rewards for correct performance (800ms, 400ms delay), a cue for the shape vs color task (200ms, 400ms delay), a trial stimulus consisting of a square or circle colored in yellow or blue (min(RT, 1400ms)), reward feedback (200ms), and an inter-trial interval (1000-1400 ms). Participants' task was to respond to the trial stimulus according to the cued rule: if the task was 'shape', respond with a key press based on the shape; if the task was 'color', respond with a key press based on the color. Participants practiced the task until they reached a 70% accuracy criterion, and then performed 10 blocks of 65 trials in the main task. Conditions were balanced within each block. For our encoding and state space analyses, we included trials 1) that were not the first trial of a block, 2) where the current and the previous trial was accurate, 3) where the current RT was longer than 200ms, and 4) that were not identified as containing artifacts during preprocessing. EEG data were recorded with 61 Ag/AgCl sintered electrodes (EasyCap; FieldTrip layout: elec1010), a NeuroScan SynAmps RT amplifier, and Curry 7 acquisition software. See (22) for full description.

3. Encoding Geometry Analysis

In our traditional multivariate analyses, we used Encoding Geometry Analysis (EGA; (30)) to quantify task encoding. At each timestep within the epoch, we fit a general linear model (GLM) to the across-trial voltage in the EEG electrodes. Our regression model included an intercept, task identity (with separate regressors for switch and repeat trials), cue identity (with seperate regressor contrasting the cues within each task), cue repetitions vs alternations for repeat trials, the main effect of switch vs repeat, the current trial's reaction time, and the previous trial's reaction time. We fit this GLM separately to even and odd runs, and then performed multivariate spatial pre-whitening on our regression estimates (69). We tested the reliability of regression weights by correlating spatial patterns across even and odd folds, and testing these correlations against zero at the group level using sign-flipping permutation tests. This analysis equivalent to out-of-sample prediction (30). To correct for multiple comparisons over time while accounting for temporal autocorrelation, we used threshold-free cluster enhancement (TFCE; (27)) with a set of EEG-optimized parameters (H=2, C=1; from (27); 10,000 permutations).

4. Temporal Generalization Analysis

To test the stability of representations over time, we used a variant of temporal generalization analysis (31) that used EGA rather than decoding. Instead of testing the reliability of encoding weights at the same timestep, we tested the similarity of weights across different timesteps (producing time × time estimates of encoding generalization). We tested the generalization against zero using sign-flipping permutation tests (TFCE-corrected; 1,000 permutations).

5. State Space Analysis

A. Generative Model. We developed a toolkit called StateSpaceAnalysis for fitting linear-Gaussian state space models (LGSSM) to neuroimaging data. Our goal was to test how effectively LGSSMs could capture rich neuroimaging data, attempting to balance the predictive power of machine learning approaches with the interpretability offered by linear-Gaussian assumptions. To provide the speed and precision needed for estimating the parameters of these large latent variable models, we developed this package using the open-source coding language Julia, available at `www.github.com/xxxx`.

The generative model for our analysis is a partially observable autoregression process: a linear dynamical system from which we can only take noisy measurements (or alternatively, that only produces noisy emissions of its underlying state). This autoregressive process consistent of a vector of latent factors ($\mathbf{x}$), and a discrete-time difference equation that describes how they evolve:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_{t-1} + \mathbf{w}_t$$

Latent factors evolve according to their recurrent dynamics ($A\mathbf{x}$), the influence of known inputs ($B\mathbf{u}$), and process noise ($\mathbf{w}_t \sim \mathcal{N}(0, W)$). At each timestep, we get a noisy observation of these latent factors:

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t$$

Observations are projections of the latent factors ($C\mathbf{x}$) corrupted by observation noise ($\mathbf{v}_t \sim \mathcal{N}(0, V)$). Note that process noise is carried forward in time through the recurrent dynamics, unlike observation noise. At the beginning of each epoch, we have an expectation about the latent state:

$$\mathbf{x}_1 = B_0\mathbf{u}_0 + \mathbf{w}_1$$

The initial conditions depend on the trial conditions ($B_0\mathbf{u}_0$) and initial uncertainty ($\mathbf{w}_1 \sim \mathcal{N}(0, W_1)$).

B. Expectation-Maximization. A major benefit of LGSSMs is that they permit a closed-form methods for Bayesian inference of the underlying latent state, as well as a closed-form marginal likelihood of the observed data , $P(\mathbf{y}_{1:T} \mid \Theta)$ where Θ represents the model parameters. This is achieved using standard inferential tools from control theory that are designed for inference over LGSSMs: Kalman filtering (inferring $\mathbf{x}_t$ from past observations) and Rauch–Tung–Striebel (RTS) smoothing (inferring $\mathbf{x}_t$ from both past and future observations).

While the marginal likelihood from the Kalman filter would allow us to directly fit our parameters though techniques like maximum likelihood estimation, in practice it is usually much more efficient to use expectation-maximization (EM; (44, 45, 70)). EM maximizes the expected lower bound of the marginal likelihood (called the ELBO) by alternating between two steps. In the E-step, we estimate the latent variables using our RTS smoother. In the M-step, we find a point-estimate of our parameters that maximize the ELBO, given our estimated latent states. With priors on our parameters, this optimizes the mode of their posterior density. The Linear-Gaussian assumptions of our model considerably simplify this procedure by allowing us to work with the sufficient statistics of our estimated latent state (i.e., their mean and covariance).

In practice, we alternate between inferring the posterior mean and covariance of the latent state using the RTS smoother (E-step), and then solving a series of regression problems to update our parameters (M-step). Alternating between these steps monotonically improves the ELBO towards a local optimum. Our toolkit is inspired by the SSM and dynamax python packages, but has several optimizations specifically for linear-Gaussian systems.

C. Subspace Identification. We found that good initialization of our parameters was critical to the effectiveness and stability of our model-fitting procedure. We initialized our parameters using a procedure called subspace identification (SSID; (71, 72)). This procedure constructs a large delay embedding matrix which concatenates lagged future and past copies of our observations ($\mathbf{y}$) and inputs ($\mathbf{u}$). The 'horizon' of these lags was set to match the largest tested model (see below). We used the canonical variate analysis procedure (73), which uses SVD to find low-dimensional mappings between lagged inputs and outputs. We can recover the A and C matrices by processing different blocks of this SVD, and then estimate the rest of the parameters using fixed-gain Kalman filtering. We performed CVA using a modified version of the `ControlSytemIdentification.jl` Julia package Method 11.

6. Fitting procedure

A. Preprocessing. We split our data into training and testing folds, preprocessing the inputs and observations of each separately. We epoched the task around the prepatory period, including the cue, the delay period, and the first 200ms of the trial (before any RTs had occured).

We used the same inputs as our sensor-level encoding model (see above). For the initial conditions, we included an intercept, the previous task, the previous reaction time, and the upcoming reaction time. We convolved our predictors with a cubic B-spline basis set. Spline knots were placed to optimize coverage at every 5 timesteps. Since we had 100 timesteps in EEG, this resulted in 20 temporal bases per predictor, tiling the entire epoch. Each input was z-scored across trials to reduce collinearity.

We preprocessed the EEG electrodes for SSA using PCA, projecting the voltage timeseries into PCs accounting for 99% of the variance. We used PCA because the observations were degenerate from spatial proximity, independent components analysis, and bad-channel interpolation. PCA also reduced the computational demands of our analyses, due to a smaller observation dimensionality and because it made our observation covariance close to diagonal. We estimated the PCs on training data, and then used these components to project the test data.

B. SSID. To provide the best initialization across our latent dimension hyper-parameter, we estimated SSID for a horizon and latent dimensionality that matched our largest tested model (128 latent factors). We then truncated these systems for smaller

models (as latent states from CVA are ordered by their singular values). We found that SSID was enhanced by reshaping our trial-wise data into one long timeseries (allowing for larger numbers of factors, and presumably low frequencies), as well as by only including inputs in the initial timesteps of each epoch (presumably reducing the collinearity between lagged inputs). Note that test data we removed before this process (which were on separate experimental blocks to minimize temporal proximity), and our EM procedure was fit within each epoch. In practice, we found that this procedure provides a good initial guess of the parameters without requiring multiple initial parameters, and was critical for stable estimation of large system (e.g., with more latent dimension that observed dimensions).

C. EM. As described above, our EM procedure cycled between estimating the latent states, and updating the parameters. We fit our model across eight different levels of latent dimensionality: $(16, 32, 48, 64, 80, 96, 112, 128)$. We set the maximum number of iterations to 20,000, measuring the test-set log-likelihood every 100 iterations. We terminated our fitting if either the total data likelihood in the training set stopped decreasing, or if the test log-likelihood stopped decreasing.

D. Parameter Recovery. We validated that our combination of model, data, and fitting procedure could produce identifiable parameter estimates when we know the generative model class. First, we used the LGSSM generative model to create a synthetic dataset based on a participant's parameters and inputs. We then performed SSA on this synthetic dataset. Next, we aligned our estimated and recovered parameters, as the estimates from SSA are only identifiable up to an invertible transformation. Intuitively, we could shuffle the ordering of the latent factors without changing the likelihood. Finally, we correlated the generating and recovered parameters (correlating the Cholesky factors for covariance matrices). While we had somewhat poorer recovery of input matrices (Figure 2C), this strongly depend on the norm of the input matrix column that we were trying to recover (i.e., some predictors were weakly encoded, and these were difficult to recover).

7. Model Comparison

A. Bayesian Model Selection. We selected the number of latent dimensions for subsequent analyses using a standard Bayesian model selection procedure (48). This analysis estimates the expected probability of each model in the population. From these posterior mixture weights, once can compute a 'protected exceedance probability', the probability that a model is the most popular within the tested set of models, relative to chance.

B. Sensor-Level Null Models. To compare SSA to a set of simpler null hypotheses, we fit a set of four models directly to the electrode PCs (i.e., without a latent embedding beyond the PCA). First was an intercept-only model (for standard R^2). Second was an encoding analysis using our full temporal basis set (a generalized additive model similar to EEG methods like the unfold toolbox, (74)). Third was a vector autoregressive (VAR(1)) model, estimating the multivariate relationship between sensor PC responses on adjacent timesteps. Fourth was a model that incorporated both encoding and autoregression. Model four, VAR with encoding, was the best-performing of these null models, so we used this as our benchmark in Figure 2D. We calculated the Cox-Snell R^2, which uses likelihoods instead of squared errors, as this metric is equivalent for normal models with stationary residual covariance (75)).

C. Brain-Optimized Recurrent Neural Networks. To compare SSA to a more complex non-linear state space model, we predicted the PC timeseries used an recurrent neural network (RNN) based autoencoder in PyTorch (76)). Each participant was fit with an RNN that matched their number of SSA parameters (at 112 latent dimensions). The inputs to this RNN were the previous PC responses and the same set of inputs as our SSA model (formatted as constant inputs over each epoch). These were linearly projected into a set of hidden units, along with the previous hidden state, and then passed through a rectified linear unit (ReLU) activation function. The PC response on the next time was linearly decoded from the hidden sate using the transpose of the observation embedding layer (a 'tied' parameterization that matches the assumptions of SSA; (77)). We found similar performance when we fit both the encoding and decoding layers, as these required more parameters. We also fit an initial hidden state, and included the first timestep in our loss.

We fit 32 random initializations of the RNN to each participant. Each fitting session involved 1000 epochs of updating the parameters by backpropagating through time the mean square error of the next-timestep predcition. Each batch contained all of the training data. Parameters were updated using the 'AdamW' learning rule (learning rate = .001, weight decay = .01; (78)). To evaluate each model's performance, we took the best test-set loss across all epochs and initializations.

8. Task State Analysis

A. Additive state decomposition. To isolate the dynamics of task encoding, we leveraged a powerful property of linear systems: the superposition principle. Linear state space models can be factorized into an additive set of subsystems for each input, a procedure known as 'additive state decomposition' (50). We used this principle to model 'task subsystems' in which state dynamics are only caused by task-related inputs. We modeled task inputs separately for switch and repeat trials, allowing

us to further decompose our system into a switch-task subsystem and a repeat-task subsystem. Within each subsystem, the state evolves accord to:

$$\mathbf{x}_t^{\text{input}} = A\mathbf{x}_{t-1}^{\text{input}} + B^{\text{task}}\mathbf{\Phi}_{t-1}$$

With $B^{\text{task}}\mathbf{\Phi}$ reflecting the estimated input, calculated from the dot product between the columns of B corresponding to a specific input (e.g., task) and our temporal basis set (i.e., the time-varying inputs for task=1). Since we contrast-coded our tasks (+1/-1), the dynamics for each task are symmetrical around the origin. For the initial conditions of this task subsystem, we used the 'previous task' component of our initial conditions, which is symmetrical between switch and repeat trials for the same task. To normalize task states across participants, we whitened the latent space by transforming our parameters by the inverse of the mean-field state noise ($\texttt{Diagonal}(W)^{-1}$).

B. Spatiotemporal Visualization. Using the decomposition and symmetry principles afforded by our linear system, we compared the estimated latent trajectories of task encoding between switch and repeat trials. We first visualized task states using group-level singular value decomposition (SVD). The goal of this visualization method was to provide an aggregate task state that accommodated the different realizations across participants (i.e., that aligned their latent states). In both of our the spatial and temporal visualizations, our use of SVD makes the sign of these visualization arbitrary.

To visualize the latent trajectories over time, we spatially concatenated participants' switch and repeat task trajectories into a wide 2D 'temporal' matrix. We projected this temporal matrix into its left singular vectors (weighted by their singular values):

$$\begin{aligned}
&\mathbf{x}_{\text{temporal}} \in \mathbb{R}^{(\text{switch timesteps}+\text{repeat timesteps})\times(\text{factors}\times\text{participants})} \\
&USV^\top = \mathbf{x}_{\text{temporal}} \\
&\mathbf{z}_{\text{temporal}} = \mathbf{x}_{\text{temporal}}VS
\end{aligned}$$

This resulted in separate component timeseries for switch and repeat subsystems. For illustration purposes, we plotted the symmetrical trajectories for each task.

C. Task Similarity Analyses. To quantify our task state visualizations, we took advantage of similar multivariate analysis tools as in our EEG analysis. We performed temporal generalization analysis on the similarity between task representations on switch and repeat subsystems by taking the cosine of the angle between task states across temporal lags. We also measured the proximity of task states across switch and repeat subsystems. We computed the Euclidean distance between either the same task across switch and repeat subsystems, or between different tasks across these subsystems.

To test whether the temporal generalization of task encoding increases over the epoch, we extracted the off-main diagonals of the similarity matrix, which reflect task similarity at different temporal lags. We averaged the diagonals above and below the main diagonal, and then correlated them with a linear time vector (i.e., tested whether the similarity at a given lag increases or decreases over timesteps). After computing these correlations for each participant, we tested for a group-level difference from zero using TFCE.

D. Task Encoding Strength. To quantify the magnitude of task representations, we used the control theoretic tools from the analysis of controllability (52, 53, 79). In control theory, the controllability quantifies the strength of of input-state coupling. For a time-invariant system, the Gramian matrix defines the asymptotic state covariance that is attributable to a (white noise) input:

$$\begin{aligned}
C_\infty &= AC_\infty A^\top + BB^\top \\
&= \sum_{t=0}^{\infty} A^t BB^\top A^{t\top}
\end{aligned}$$

To capture the influence of a particular set of time-varying inputs, we simulated the trajectory of this Gramian using a matrix-variate difference equation:

$$\begin{aligned}
G_t &= AG_{t-1}A + (B^{\text{task}}\mathbf{\Phi}_{t-1})(B^{\text{task}}\mathbf{\Phi}_{t-1})^\top \\
G_1 &= (B_0^{\text{prevTask}})(B_0^{\text{prevTask}})^\top
\end{aligned}$$

Here, G reflect the task-dependent Gramian, equivalent to the covariance for white noise inputs. It is similar in spirit to the 2-norm of the task state (41), but takes into account the full covariance across factors. We summarized task strength using a standard metric of 'average controllability' (52), taking the trace of G each timestep (i.e., the sum of G's eigenvalues).

E. Model Ablation. We eliminated different components of our fitted model to assess their contributions to our task state trajectories. We assessed the role of system recurrence by setting the recurrence matrix (A) and initial conditions ($B_0 u_0^{\text{PreviousTask}}$) to zero. We assessed the role of input strength by normalizing Bu_t^{task} at each timestep (dividing by the column 2-norm). We found similar results when we excluded early task inputs in the recurrence ablation, or when we normalized within each state for the input ablation (i.e., normalizing rows of Bu_t^{task}).

F. Performance Alignment. To test whether our inferred task states mattered for good performance, we explored how trial-level brain states predicted upcoming reaction times. First, we projected each participants' task trajectory for switch and repeat subsystems back into the observation space ($C\mathbf{x}^{\text{task}}$). On every trial and timestep, we computed the cosine similarity between the task state and the brain response. We then put this similarity into a regression along with predictors accounting for task, switch trials, task $\times$ switch interaction, and the 2-norm of the neural response. We tested whether EEG-task alignment was significantly different from zero at the group level using TFCE.

9. Task-Optimized RNNs

A. Task. We trained networks to perform an extension of the 'context-dependent decision-making' task (CDM; (60–62, 80), which was designed to be highly similar to the task that participants performed during EEG.

We trained networks on two kinds of sequences. On 'single sequences', networks performed a classic CDM task. On each sequence, networks first got a task cue input (one-hot coded; 10 timesteps), followed by a delay period (20 timesteps). They then got two pairs of stimulus inputs (40 timesteps), having to judge which input in the task-relevant pair had higher amplitude. We trained the networks using a logistic loss function on the correct and incorrect response options ($-[y\log(\hat{y})+(1-y)\log(1-\hat{y})]$).

On 'double sequences', networks performed two consecutive trials (i.e., without resetting the network or gradients). They performed the first trial, had a short inter-trial interval (20 timesteps) and then performed the second trial. Sequences within an epoch had a equal number of target repetitions, distractor repetitions, and task repetitions (64 conditions).

Each epoch contained 150 conditions sets (9600 sequences), each with normally distributed noise (cue d' = 1.0, stimulus d' = 0.15). We trained at relatively low SNRs to pressure the networks to find task-relevant states.

B. Network Architecture. We trained RNNs with gated recurrent units (GRUs; 108 hidden units; (81)) to perform task-switching. These networks had a standard RNN architecture: linearly encoded inputs, a recurrent hidden state with hyperbolic tangent activation function, and linear decoded outputs. The GRU component of these models learn a pair of 'gates', each of which encode the hidden state and inputs, and output a sigmoid that is element-wise multiplied with hidden states. The 'reset gate' is applied to the hidden state before it is combined with the inputs and passed through the activation function to produce the 'proposal state'. The update gate mixes the proposal state and the previous hidden state. We trained GRUs using backpropagation through time, using the AdamW learning rule (learning rate = .01, weight decay = .01).

C. Curricula. We trained two groups of GRUs on different curricula. 'Single networks' only trained on the single sequences (500 epochs). 'Mixed networks' trained on single sequences for 90% of their training (450 epochs), and then double sequences for the final 10% (50 epochs). The rationale for this design was to teach both groups how to perform the task, but give the Mixed networks enough experience that they would learn a compensatory strategy for switching. Note that 450 epoch constituted over-training on the task, at a point where the 'test loss' (noise-free loss) was increasing (Figure S6). We trained 512 networks in each group, using matched random seed to ensure that training was the same for the first 90% of epochs.

D. Performance Evaluation. Networks were tested on noise-free double sequences to evaluate whether they had learned the underlying task structure and could switch between tasks. We measured the within-sequence performance through the per-timestep loss, averaged over conditions and log transformed. To test for temporal generalization, we evaluated networks on 'triple sequences', which included a third trial that was absent from training.

E. Task Generalization. To test the generality of mixed networks' capacity to switch between task, we explored a curriculum for assessing abstract flexibility. In this curriculum, single sequences involved learning four different tasks (mapped to two responses). In double sequences, networks switched and repeated only two out of the four tasks. Single networks performed 1000 epochs of single sequences, and mixed networks performed 900 single and 100 double. We then assessed both networks' switching ability on double sequences containing the two tasks that had been held out of switching practice, comparing the timestep-resolved loss on switch and repeat trials.

F. Gate Ablations. To assess the role of the reset and update gates in GRU's performance, we trained a series of model ablations. We either fixed the reset gate to be open, fixed the update gate to be open, or trained a vanilla RNN. GRUs were

trained with tanh activation function and learning rate = .01, whereas the RNN was trained with a ReLU activation function and learning rate = .001 (which improved its performance). Free parameters were matched across networks (108 hidden units for full model, 134 units for ablated models, 190 units for RNN). We trained these models on 500 epochs of the single sequences, and then 500 epochs of the double sequences.

G. Hidden Unit Visualization. We used a similar group SVD procedure to our EEG (see Method 8) to visualize hidden unit activation during double sequences. Using trained networks, we first simulated a set of noisy trials (n=2048) for a single and mixed network (same random seed). Next, we concatenated the hidden unit activations across sequences and networks into a joint matrix $((\text{timesteps} \times \text{sequences}) \times (\text{ single-network hidden units} + \text{ mixed-network hidden units}))$ Performing SVD on this matrix provides embeddings that captures the common and distinct components of the hidden unit activation, and in practice it produces quite similar trajectories across the two networks. We then averaged the sequences within each task pair and projected these average sequences into the (weighted) right singular vectors corresponding to each network.

10. Network Distillation

We used our SSA package to distill trained RNNs into a high-dimensional state space model, allowing us to compare the task dynamics to human participants. Our fitting procedure was closely matched between networks and EEG. We preprocessed training and test trials in a similar way (1536 trials for training, 512 for testing): performing PCA on the hidden unit trajectories (keeping components accounting for 99% of the variance) and constructing a temporal basis for the inputs (10 spline bases over the epoch). Network inputs were a bias, task on switch trials, task on repeat trials, and the main effect of switch. Inputs to initial conditions were a bias and the previous task. We analyzed the preparation preparation for the second trial: cue period (10 timesteps), delay period (20 timesteps), and the initial trial period (10 timesteps).

We fit SSA across five levels of latent dimensionality $(16, 40, 64, 88, 112)$, initializing our parameters with SSID at a horizon of 112. We ran SSA for all 512 single networks and mixed networks. Networks were evaluated using the same methods as human participants.

To evaluate the similarity between temporal generalization matrices in networks and humans, we subsampled human task trajectories to match the duration of networks' trajectories, averaged the temporal generalization within humans and RNNs, and then computed the cosine similarity between these (vectorized) matrices. Note that we found qualitatively similar results evaluating similarities on a per-network basis.

11. Software and Visualization

1. Programming Languages
   (a) Julia (v1.10.4)
   (b) Python (v3.10.13)
   (c) Matlab (R2023b)
2. Software Packages
   (a) PyTorch (v2.4.0; pytorch.org)
   (b) MatlabTFCE (github.com/markallenthornton/MatlabTFCE)
   (c) ControlSystemIdentification.jl (v2.10.2; github.com/baggepinnen/ControlSystemIdentification.jl)
   (d) FieldTrip (82)
   (e) EEGLAB (v2024.2; (83))
3. Visualization
   (a) HenriquesLab bioRxiv template (overleaf.com/latex/templates/henriqueslab-biorxiv-template/nyprsybwffws)
   (b) Scientific Color Maps (84)

The author(s) are pleased to acknowledge that the work reported on in this paper was substantially performed using the Princeton Research Computing resources at Princeton University which is consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing.

12. Supplementary Figures

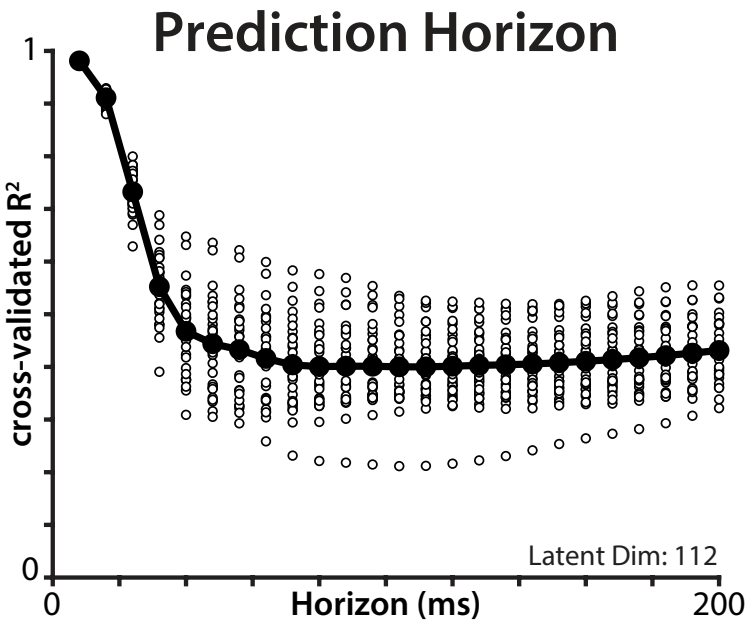

Fig. S1. SSA prediction horizon. Predictive accuracy of the SSA model when forecasting at different timesteps into the future. White dots reflect individual participants.

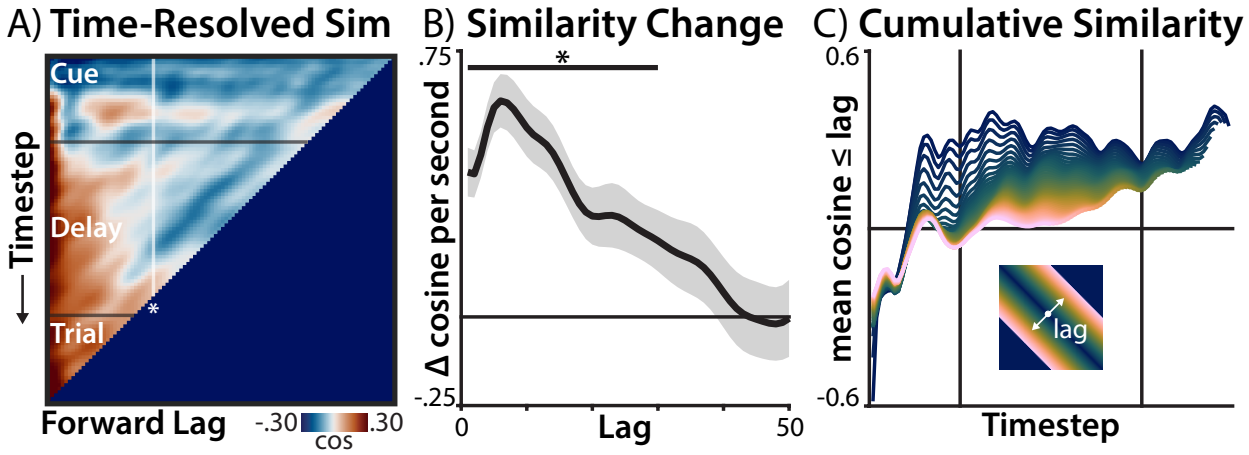

Fig. S2. Change in generalization over the epoch. **A)** Average cosine similarity between task patterns across switch and repeat conditions, plotted at every timestep (y-axis) and time lag (x-axis;compare to Figure 3C). Vertical white line reflect significance cut-off from (B). **B)** Change in similarity over the epoch (y-axis; linear regression estimate), as a function of lag (x-axis). Similarity change is significantly greater than zero for lags up 31 timesteps (248 ms; TFCE-corrected). Error bars reflect SEM. Corresponds to columns of (A). **C)** At each timestep (x-axis), the cumulative similarity up to each lag (line color; see inset for legend). Corresponds to rows of (A).

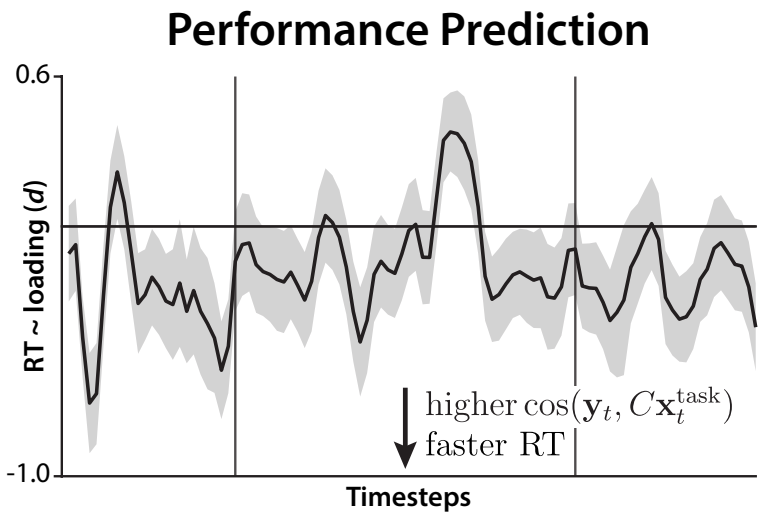

Fig. S3. Relevance of task states for good performance. Group-level effect size (y-axis; Cohen's d) of the individual regression weights mapping 'task loading' on to log reaction time. Task loading is the cosine similarity between EEG activity and the estimated task state. Error bars reflect SEM. No individual timestep is significantly different from zero (using TFCE), however the sum of the regression weights over time is significant ($d = -0.53,$ t-test $p = .0068$), with stronger alignment predicting shorter reaction times.

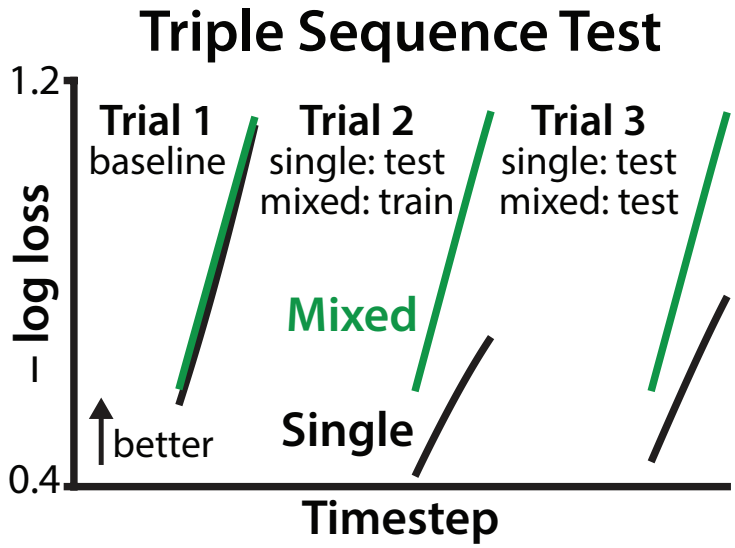

Fig. S4. Sequence generalization. Average test loss across timesteps within a 'triple sequence'. Networks were trained on single sequences (single networks) or single and double sequences (mixed networks), and then tested on this three-trial sequence. SEM smaller than line thickness.

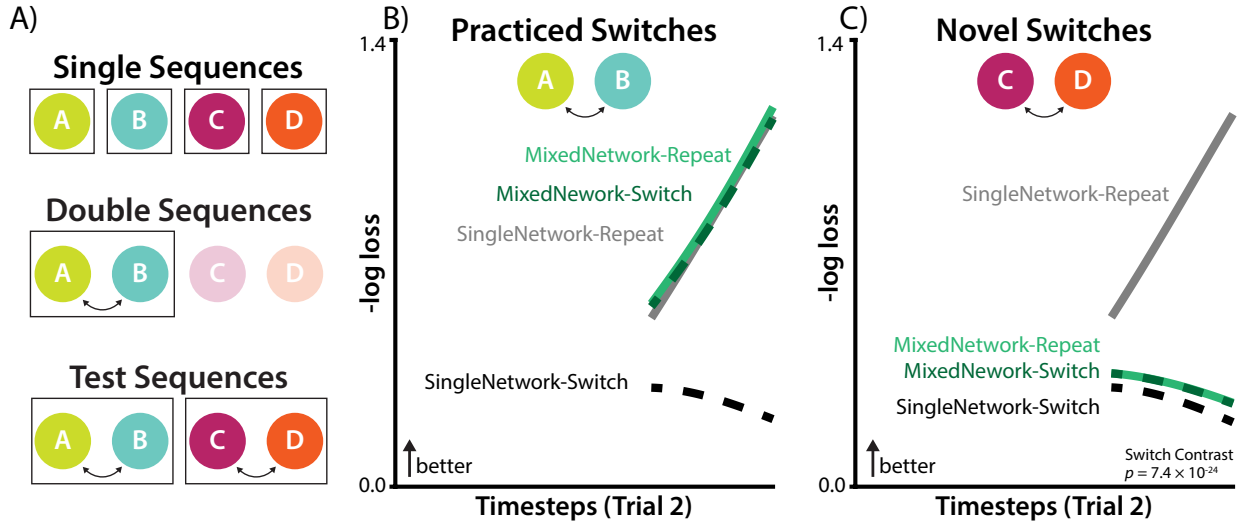

Fig. S5. Task generalization. **A)** We trained GRUs on a modified version of our protocol. In single sequences, networks learned four different tasks. In double sequences, networks practiced switching between two of the tasks. In test (shown in A and B), we examined switching performance between the practiced and novel task pairs. As in our main experiment, single networks performed only single sequences (1000 epochs), and mixed networks were trained on 90% single sequences and 10% double sequences. **B)** The mixed network shows great performance on task transitions that it had practiced, it (as good at switching as the single network is at repeating). This is similar to the results Figure 4F (shown fore each condition, rather than as contrasts). **C)** On novel task transitions, the mixed network still shows lower switch costs and better performance on switch trials. Note that it has worse performance on repeat trials, likely due to these networks having less experience with C and D tasks (and/or catastrophic interference)

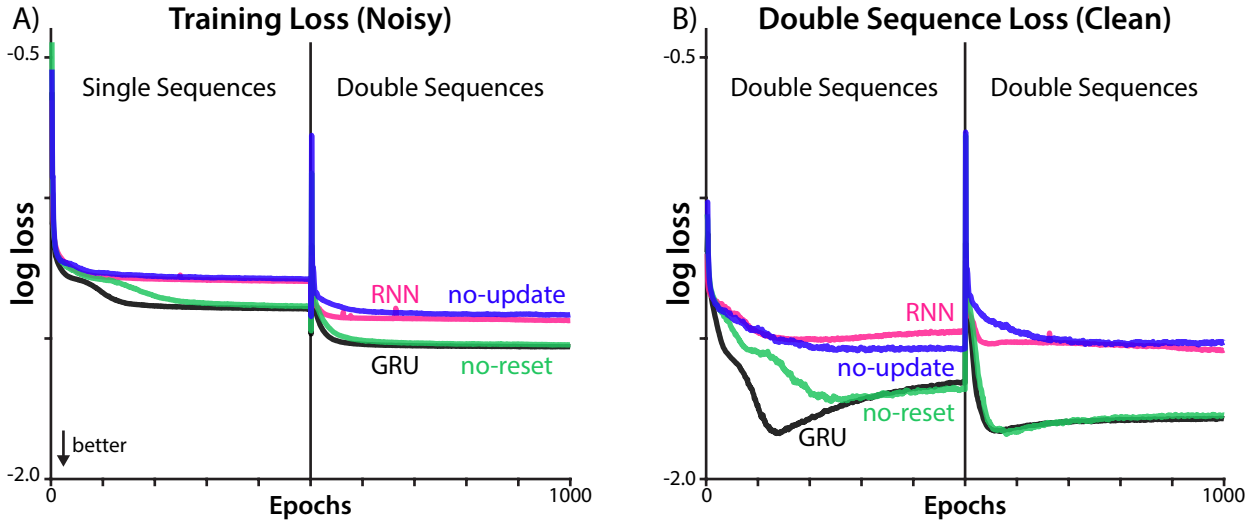

Fig. S6. Gate Ablation. We trained models with all gates ('GRU', black), fixed-open reset gates ('no-reset', green), fixed-open update gates ('no-update', blue), and classic Elman RNNs ('RNN', pink). All models are matched for the number of parameters (see Methods). The first epoch is removed to improve visualization. Note that we plot loss (instead of -loss) to follow convention. **A)** We train networks on single sequences for 500 epochs, and then double sequences for 500 epochs. During this training, noise was added to inputs to encourage the learning of a robust policy. During learning, GRU and no-reset have similar learning trajectories, with better performance (lower loss) than no-update or RNN models. **B)** Network loss was also tested on double sequences at every epoch, without input noise. Here too, the GRU and no-reset show much stronger gains in performance from experience with the double sequences. Note that GRU test loss during the single sequences is increasing, suggesting that this is in an over-trained regime.

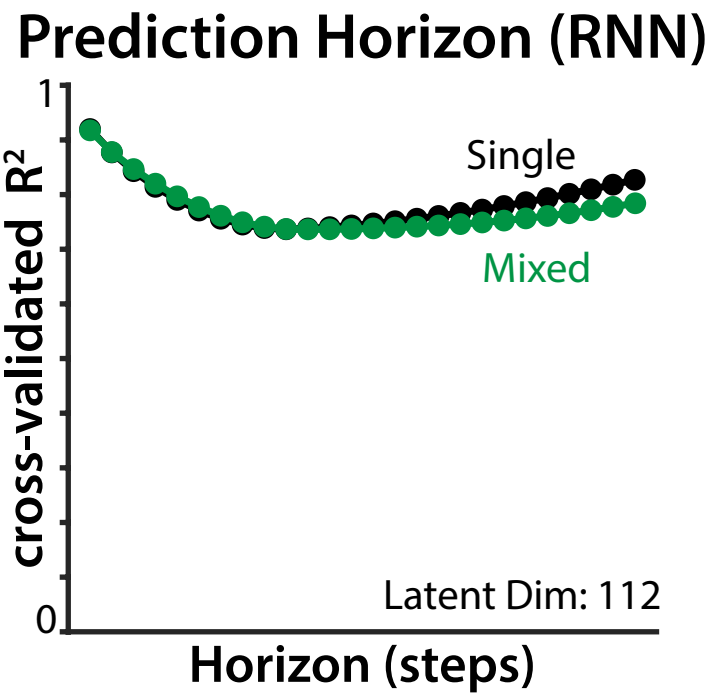

Fig. S7. Prediction horizon for SSAs fit to RNNs. SSA had similarly good predictive accuracy when fit to either single or mixed networks had. This high predictive accuracy was present both at the next timestep (training objective) and at distant temporal lags (25 timesteps is more than half of the epoch).