Flexible task abstractions emerge in linear networks with fast and bounded units

Kai Sandbrink[*]
Exp. Psychology, Oxford
Brain Mind Institute, EPFL

Jan P. Bauer[*]
ELSC, HebrewU
Gatsby Unit, UCL

Alexandra M. Proca[*]
Department of Computing
Imperial College London

Andrew M. Saxe
Gatsby Unit, UCL

Christopher Summerfield
Exp. Psychology, Oxford

Ali Hummos[*†]
Brain and Cognitive Sciences
MIT

Abstract

Animals survive in dynamic environments changing at arbitrary timescales, but such data distribution shifts are a challenge to neural networks. To adapt to change, neural systems may change a large number of parameters, which is a slow process involving *forgetting* past information. In contrast, animals leverage distribution changes to segment their stream of experience into tasks and associate them with internal task abstracts. Animals can then respond *flexibly* by selecting the appropriate task abstraction. However, how such flexible task abstractions may arise in neural systems remains unknown. Here, we analyze a linear gated network where the weights and gates are jointly optimized via gradient descent, but with neuron-like constraints on the gates including a faster timescale, nonnegativity, and bounded activity. We observe that the weights self-organize into modules specialized for tasks or sub-tasks encountered, while the gates layer forms unique representations that switch the appropriate weight modules (task abstractions). We analytically reduce the learning dynamics to an effective eigenspace, revealing a virtuous cycle: fast adapting gates drive weight specialization by protecting previous knowledge, while weight specialization in turn increases the update rate of the gating layer. Task switching in the gating layer accelerates as a function of curriculum block size and task training, mirroring key findings in cognitive neuroscience. We show that the discovered task abstractions support generalization through both task and subtask composition, and we extend our findings to a non-linear network switching between two tasks. Overall, our work offers a theory of cognitive flexibility in animals as arising from joint gradient descent on synaptic and neural gating in a neural network architecture.

1 Introduction

Humans and other animals show a remarkable capacity for flexible and adaptive behavior in the face of changes in the environment. Brains leverage change to discover latent factors underlying their sensory experience [Gershman and Niv, 2010, Yu et al., 2021, Castañón et al., 2021]: they segment the computations to be learned into discrete units or 'tasks'. After learning multiple tasks, low-dimensional task representations emerge that are abstract (represent the computation carried invariant to the current sensory content) [Bernardi et al., 2020] and compositional [Tafazoli et al., 2024]. The discovery of these useful task abstractions relies on the temporal experience of change,

[*]Equal contribution, randomized order
[†]Corresponding author, ahummos@MIT.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

and in fact, brains struggle when trained on randomly shuffled interleaved data [Flesch et al., 2018, Beukers et al., 2024].

In contrast, while artificial neural networks have become important models of cognition, they perform well in environments with large, shuffled datasets but struggle with temporally correlated data and distribution shifts. To adapt to changing data distributions (or 'tasks'), neural networks rely on updating their high-dimensional parameter space, even when revisiting previously learned tasks – leading to catastrophic forgetting [McCloskey and Cohen, 1989, Hadsell et al., 2020]. One way to limit this forgetting is through task abstractions, either provided to the models [Hummos et al., 2024] or discovered from data [Hummos, 2023]. In addition, adapting a model entirely by updating its weights is data-intensive due to high dimensionality. Task abstractions simplify this process by allowing updates to a low-dimensional set of parameters, which can be switched rapidly between known tasks, and recomposed for new ones. However, despite the advantages of task abstractions, simple algorithms for segmenting tasks from a stream of data in neural systems remain an open challenge.

This paper examines a novel setting where task abstractions emerge in a linear gated network model with several neural pathways, each gated by a corresponding gating variable. We jointly optimize the weight layer and gating layer through gradient descent, but impose faster timescale, nonnegativity, and bounded activity on the gating layer units making those parameters closer conceptually to neurons. We find two discrete learning regimes for such networks based on hyperparameters, a *flexible* learning regime in which knowledge is preserved and task structure is integrated flexibly, and a *forgetful* learning regime in which knowledge is overwritten in each successive task. In the flexible regime, the gating layer units align to represent tasks and subtasks encountered while the weights separate into modules that align with the computations required. Later on, gradient descent dynamics in the gating layer neurons can retrieve or combine existing representations to switch between previous tasks or solve new ones. Such flexible gating-based adaptation offers a parsimonious mechanism for continual learning and compositional generalization [Butz et al., 2019, Hummos, 2023, Qihong Lu et al., 2024, Schug et al., 2024]. Our key contributions thus are as follows:

- We **describe flexible and forgetful modes of task-switching** behavior that arise in neural networks and **analytically identify the effective dynamics** that induce the flexible regime.
- The model, to our knowledge, is **the first simple neural network model that benefits from data distribution shifts and longer task blocks rather than interleaved training** [Flesch et al., 2018, Beukers et al., 2024]. We also provide a direct comparison to human behavior where task switching accelerates with further task practice [Steyvers et al., 2019].
- **We generalize our findings to fully-connected deep linear networks**. We find that differential learning rates and regularization on the second layer weights are necessary and sufficient for earlier layers to form task-relevant modules and later layers to implement a gating-based solution that selects the relevant modules for each task.
- **We extend our findings to non-linear networks.** As a limited proof of concept, we embed such a layer in a non-linear convolutional network learning two-digit classification tasks.

2 Related work

Cognitive flexibility is a set of abilities allowing brains to adapt behavior in response to change [Miller and Cohen, 2001, Egner, 2023, Sandbrink and Summerfield, 2024]. Neural network models of cognitive flexibility frequently represent knowledge for different tasks in distinct neural populations, or modules, which then need to be additionally gated or combined [Musslick and Cohen, 2021, Yang et al., 2019]. Several models assumed access to ground truth task identifiers and used them to provide information about the current task demands to the network [Kirkpatrick et al., 2017, Masse et al., 2018, Yang et al., 2019, Wang and Zhang, 2022, Driscoll et al., 2024, Hummos et al., 2024]. Indeed having access to task identifiers facilitates learning, decreases forgetting, and enables compositional generalization [Yang et al., 2019, Masse et al., 2022, Hummos et al., 2024]. Such works sidestep the problem of discovering these task representations from the data stream.

Other models train modular neural structures end-to-end, such as mixture-of-experts [Jacobs et al., 1991, Jordan and Jacobs, 1994, Tsuda et al., 2020], or modular networks [Andreas et al., 2016, Kirsch et al., 2018, Goyal et al., 2019]. A fundamental issue is the 'identification problem' where different

assignments of experts to tasks do not significantly influence how well the model can fit the data, making identification of useful sub-computations via specialized experts difficult [Geweke, 2007]. Practically, this results in a lack of modularity with tasks learned across many experts [Mittal et al., 2020] or expert collapse, where few experts are utilized [Krishnamurthy et al., 2023]. Recent work used a surprise signal to allow temporal experience to adapt learning [Barry and Gerstner, 2022]. Our model proposes simple dynamics that benefit from the temporal structure to assign sub-tasks to modules.

Our work builds on the theoretical study of *linear* networks which exhibit complex learning dynamics, but are analytically tractable [Saxe et al., 2013, 2019]. Prior work examined how gating alleviates interference [Saxe et al., 2022], but gating was static and provided as data to the network. We generalize this line of work by showing how appropriate gating emerges dynamically. More recently, Shi et al. [2022] analyzed specialization of a linear network with multiple paths when tasks are presented without blocking and gates, and Lee et al. [2024] studied the effects of a pretraining period. We consider continual learning with a blocked curriculum. Schug et al. [2024] proved that learning a linear number of (connected) task module combinations is sufficient for compositional generalization to an exponential number of module combinations in a modular architecture similar to ours. Instead, we explicitly study the interaction between task learning and gating variable update dynamics.

3 Approach

3.1 Setting

We formulate a dynamic learning problem consisting of M distinct tasks. At each time step t the network is presented with an input and output pair $(\boldsymbol{x}(t), \boldsymbol{y}^{\star m}(\boldsymbol{x}(t)))$ sampled from the current task m. Tasks are presented in blocks lasting a period of time τ_B before switching to another task sequentially (Fig. 1A). Models are never given the task identity m or task boundaries.

Specifically, we consider a multitask teacher-student setup in which each task is defined by a teacher $\boldsymbol{W}^{\star m}$, which generates a ground truth label $\boldsymbol{y}^{\star m} = \boldsymbol{W}^{\star m}\boldsymbol{x}$ with a Gaussian i.i.d. input $\boldsymbol{x}$ at every point in time. We randomly generate the teachers to produce orthogonal responses to the same input. While orthogonal tasks simplify theoretical analysis, we generalize to non-orthogonal tasks in Appendix A.9.

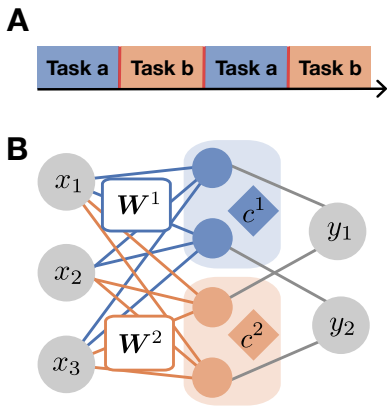

Figure 1: The open-ended learning setting and the modeling approach. **A.** Example of the blocked curriculum with two tasks. **B.** Neural Task Abstraction (NTA) model updates $\boldsymbol{W}^p$ through gradient descent, but also the gating variables c^p, leading to task abstractions emerging in the gating layer.

3.2 Model

We study the ability of linear gated neural networks [Saxe et al., 2013, 2022] to adapt to teachers sequentially. We use a one-layer linear gated network with P student weight matrices $\boldsymbol{W}^p \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, together with P scalar variables $c^p \in \mathbb{R}$ which gate a cluster of neurons in the hidden layer (Fig. 1B).

The model output $\boldsymbol{y} \in \mathbb{R}^{d_{\text{out}}}$ reads

$$\boldsymbol{y} = \sum_{p=1}^{P} c^p \boldsymbol{W}^p \boldsymbol{x}. \qquad \text{(1, NTA)}$$

Since the c^p variables will learn to reflect which task is currently active, we refer to their activation patterns as *task abstractions*. We refer to a student weight matrix together with its corresponding gating variable as a *path*.

We refer to this architecture as the Neural Task Abstractions (NTA) model when the following two conditions are met during training: first, we update both the weights $\boldsymbol{W}^p$ and the gating variables c^p via gradient descent, but on a regularized loss function $\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}}$. Second, we impose a shorter timescale for the gates τ_c than for the weights τ_w, i.e. $\tau_c < \tau_w$ (although this condition becomes unnecessary if the task is sufficiently high-dimensional, see Appendix A.2).

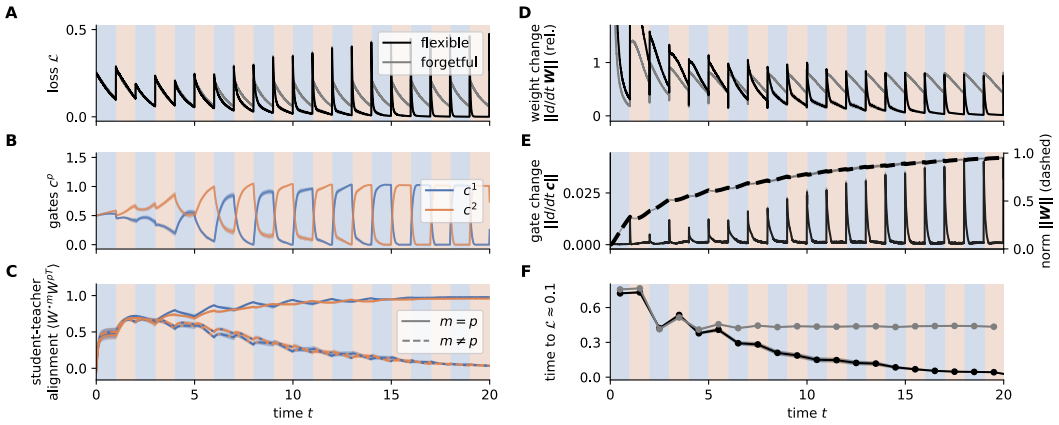

Figure 2: Joint gradient descent on gates and weights enables fast adaptation through gradual specialization. Learning on the blocked curriculum from Fig. 1 with $\tau_c = 0.03$, $\tau_w = 1.3$, and block length $\tau_B = 1.0$. x-axis indicates time as multiples of τ_B. (*Black*) Flexible NTA model Eq. (1, NTA), (*gray*) forgetful NTA model with $\tau_c = \tau_w$ and $\lambda_{\text{nonneg}} = \lambda_{\text{norm}} = 0$. Simulation averaged over 10 random seeds with standard error indicated. **A.** Loss of both models over time. **B.** Gate activity of flexible NTA. **C.** Student-teacher weight alignment $\boldsymbol{W}^{\star m}\boldsymbol{W}^{p\intercal}$, normalized and averaged over rows (cosine similarity) for each student-teacher pair. **D., E.** Norm of updates to $\boldsymbol{W}^p$ and $\boldsymbol{c}$. *Dashed*: norm of students correlating with update size of $\boldsymbol{c}$. **F.** Time to $\mathcal{L}_{\text{task}} = 0.1$ for both models over blocks.

The task loss is a mean-squared error $\mathcal{L}_{\text{task}} = {}^1\!/_2 \sum_i^{d_{\text{out}}} \langle (y_i^{\star m} - y_i)^2 \rangle$ where the average is taken over a batch of samples. The regularization loss contains two components $\mathcal{L}_{\text{reg}} = \lambda_{\text{norm}} \mathcal{L}_{\text{norm}} + \lambda_{\text{nonneg}} \mathcal{L}_{\text{nonneg}}$ weighted by coefficients $\lambda_{\text{norm}}, \lambda_{\text{nonneg}}$. The normalization term bounds gate activity, $\mathcal{L}_{\text{norm}} = {}^1\!/_2 \left(||\boldsymbol{c}||_k - 1\right)^2$, where we consider $k = 1, 2$. The nonnegativity term favors positive gates $\mathcal{L}_{\text{nonneg}} = \sum_{p=1}^{P} \max(0, -c^p)$. Together, these regularizers incentivize the model to function as an approximate mixture model by driving solutions towards any convex combination of different students without favoring specialization (see Appendix B.3 for details).

Assuming small learning rates τ_c^{-1}, τ_w^{-1} (*gradient flow*), this approach implies updates of

$$\tau_c \, {}^d\!/_{dt}\, c^p = -\nabla_{c^p} \mathcal{L}, \qquad \tau_w \, {}^d\!/_{dt}\, \boldsymbol{W}^p = -\nabla_{\boldsymbol{W}^p} \mathcal{L}$$

where τ_c and τ_w are time constants of the model parameters. We initialize $\boldsymbol{W}^p$ as i.i.d. Gaussian with small variance σ^2 / d_{in}, $\sigma = 0.01$ and $c^p = {}^1\!/_2$.

Code for model and simulations at: https://github.com/aproca/neural_task_abstraction

4 Task abstractions emerge through joint gradient descent

We train the model with fast and bounded gates on $M = 2$ alternating tasks (Fig. 1A) and use two paths $P = 2$ for simplicity (for the $P \lesseqgtr M$ case, see Fig. 3 and Appendix A.3). As a baseline, we compare to the same model but without gate regularization and timescales difference.

Both models reach low loss in early blocks, but only flexible NTA starts to adapt to task switches increasingly fast after several block changes (Fig. 2A,F). Analyzing the model components reveals what underlies this accelerated adaptation (Fig. 2C,D): in early blocks of training, zero loss is reached through re-aligning both students $\boldsymbol{W}^p$ to the active teacher $\boldsymbol{W}^{\star m}$ in every block, while the gates c^p are mostly drifting (Fig. 2B). Reaching low loss is furthermore only achieved towards the end of a block. Later, the weights stabilize to each align with one of the teachers (Fig. 2C,D), and the appropriate student is selected via gate changes (Fig. 2B), reducing loss quickly. The rate at which gates change correlates with the alignment and magnitude $||\boldsymbol{W}^p||$ of the learned weights (Fig. 2C,E). Overall, this points towards a transition between two learning regimes: first, learning happens by aligning student weight matrices with the current teacher, which we call *forgetful* because it overwrites previous weight changes. Later, as the weights specialize, the learned representations $\boldsymbol{W}^p$ can be rapidly selected by the gates according to the task at hand, reflecting adaptation that is *flexible*. Only

the model equipped with fast and bounded gates (flexible NTA) is able to enter this flexible regime (Fig. 2A,F).

Next, we verify that the task abstractions in the gating variables are general, in the sense that they support compositional generalization. We consider two settings that begin by training a model with three paths on three teachers A, B, and C in alternating blocks, and then training on novel conditions. In **task composition**, the novel conditions are the teachers' additive compositions A+B, A+C, B+C (Fig. 3A), we see that the flexible NTA model trains on these combinations much faster (Fig. 3C). In **subtask composition**, the novel conditions are combinations of the rows of different teachers, i.e. we break the teachers A, B, C into rows and select from these rows to compose new tasks. (Fig. 3B). In the subtask composition case, we use a more expressive form of the gates in the model that can control each row of the student matrices $\mathbf{W}^p$ individually. We find that, in the flexible regime, the model quickly adapts to the introduction of compositional tasks, while the forgetful model with regularization removed does not (Fig. 3C,D). For more details and extended analysis, see Appendix A.8.

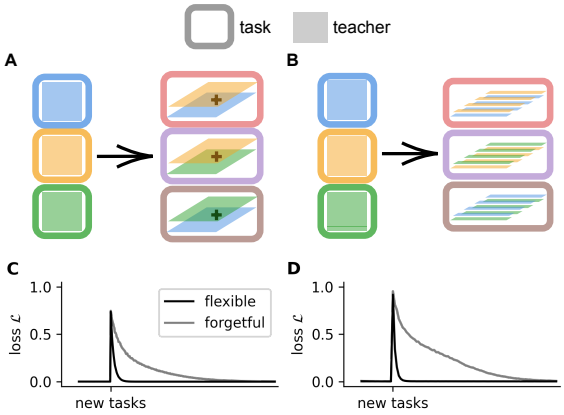

Figure 3: Flexible model generalizes to compositional tasks. **A.** Task composition consists of new tasks that sum sets of teachers previously encountered. **B.** Subtask composition consists of new tasks that concatenate alternating rows of sets of teachers previously encountered. Loss of models trained on generalization to task composition (**C.**) and subtask composition (**D.**) for the flexible (*black*) and forgetful (*gray*) NTA. 'New tasks' indicates the start of the generalization phase when the task curriculum is changed to cycle through the compositional tasks.

We devote the next section to identifying what factors support the flexible regime of learning.

5 Mechanisms of learning flexible task abstractions

We observed in Fig. 2 that simultaneous gradient descent on weights and gates converges to a flexible regime capable of rapid adaptation to task changes. But what mechanisms facilitate this solution? We here leverage the linearity of the model to identify the effective dynamics in the SVD space of the teachers, in which we describe the emergence and functioning of the flexible regime.

5.1 Reduction to effective 2D model dynamics

For simplicity, we consider the case with only $M = P = 2$ teachers and students. We take a similar approach to Saxe et al. [2013], and project the student weights into the singular value space of the teachers for each mode α individually, yielding a scalar $w^p_{m\alpha} := \boldsymbol{u}^{\star m\intercal}_\alpha \boldsymbol{W}^p \boldsymbol{v}^{\star m}_\alpha$. Each pair of components α thus reduces to a 2D state $\boldsymbol{y} = c^1\boldsymbol{w}^1 + c^2\boldsymbol{w}^2$, where we stack w^p_m into a vector along the index m and omit α in the following for readability (Fig. 4A). A similar projection is possible in terms of the row vectors of both teachers (Appendix A.1.1).

The essential learning dynamics of the system can therefore be described as

$$\tau_w \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{w}^p = c^p\left(\boldsymbol{y}^{\star m} - \boldsymbol{y}\right), \qquad (1) \qquad \tau_c \frac{\mathrm{d}}{\mathrm{d}t}c^p = \boldsymbol{w}^{p\intercal}\left(\boldsymbol{y}^{\star m} - \boldsymbol{y}\right) - \lambda \nabla_{c^p}\mathcal{L}_{\text{reg}}. \qquad (2)$$

where $\boldsymbol{y}^{\star m}$ describes the output of the currently-active teacher m. In Appendix A.1, we show analytically and through simulations that this reduction is exact when gradients are calculated over many samples.

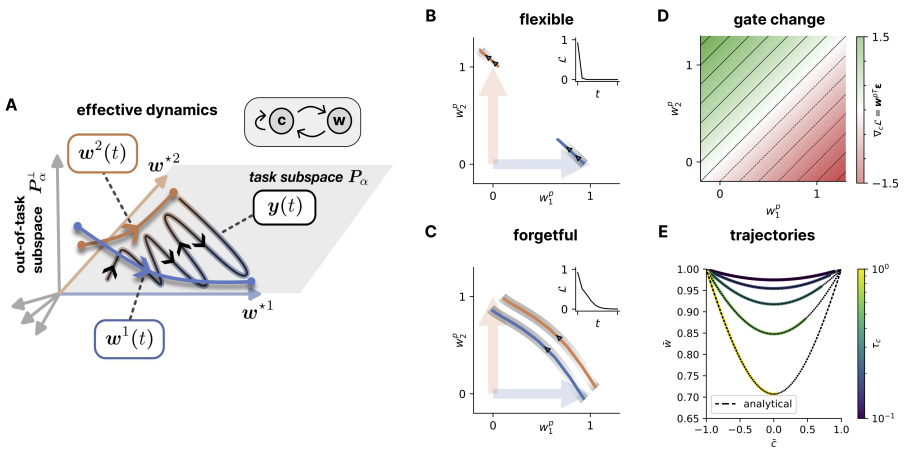

Figure 4: Mechanism of gradual task specialization in effective 2D subspace. A. Sketch of the reduced model and dynamic feedback. Out-of-subspace students gradually align to teacher axes. **B.** Trajectories of student weight matrices (*blue, orange*) in the teacher subspace during complete adaptation following a context switch from teacher 1 to teacher 2 in the flexible regime. *Gray stripes* indicate associated gate activation. The student weight matrices move little. **C.** Like (**B**), but for the forgetful regime. Student weight matrices entirely remap and gates do not turn off. **D.** Gradient of the task loss on c^p as a function of the weight alignment. **E.** Trajectories in the specialization subspace as a function of gate timescale for values $\tau_c = 0.1, 0.18, 0.32, 0.56, 1.00$ comparing (*color*) simulations and (*dashed black*) analytical predictions from exact solutions under symmetry in the flexible regime. Simulations begin from initial conditions of complete specialization and separation $w_m^p = \delta_{pm}$, $c^p = \delta_{p1}$ and follow a complete adaptation from teacher 1 to teacher 2 over the course of a block, reaching $\mathcal{L}_{\text{task}} < 10^{-2}$ for all τ_c.

5.2 Specialization emerges due to self-reinforcing feedback loops

The flexible regime is characterized by students that are each attuned to a single teacher (Fig. 4B), whereas in the forgetful regime, both students track the active teacher together (Fig. 4C). We can describe this difference by studying the specialization of the students. We define this by considering the difference in how represented the teachers are in the two paths: for teacher 1, $\bar{w}_1 := w_{m=1}^{p=1} - w_{m=1}^{p=2}$ and, for teacher 2, $\bar{w}_2 := w_{m=2}^{p=2} - w_{m=2}^{p=1}$. Similarly, a hallmark of the flexible regime are separated gates. Together, this defines the specialization subspace

$$\bar{w} := \frac{1}{2}(\bar{w}_1 + \bar{w}_2), \qquad (3) \qquad\qquad \bar{c} := c^1 - c^2 \qquad (4)$$

The system is in the flexible regime when absolute values of $\bar{w}$ and $\bar{c}$ are high (approaching 1), and in the forgetful regime when they are low (around 0). In this section, we study the emergence of the flexible regime through self-reinforcing feedback loops, with specialized students and normalizing regularization leading to more separated gates, and separated gates in turn leading to more specialized students. In each subsection, we first describe the effect of the feedback loops on the paths individually, before considering the combined effect on specialization. Without loss of generality, we consider cases where the student p specializes for teacher m.

5.2.1 Specialized students and regularization encourage fast and separated gates

We first investigate the influence of $\boldsymbol{w}^p$ on $\frac{\mathrm{d}}{\mathrm{d}t}c^p$. From the gate update in Eq. (2), we get

$$\tau_c \frac{\mathrm{d}}{\mathrm{d}t} c^p = \varepsilon_1 \, \boldsymbol{w}^{p\mathsf{T}} \boldsymbol{w}^{\star 1} \; + \; \varepsilon_2 \, \boldsymbol{w}^{p\mathsf{T}} \boldsymbol{w}^{\star 2} \quad - \quad \nabla_{c^p} \mathcal{L}_{\text{reg}}, \qquad (5)$$

where we decomposed the error $\boldsymbol{\varepsilon} := \boldsymbol{y}^{\star m} - \boldsymbol{y}$ into the teacher basis as coefficients $\varepsilon_m := \boldsymbol{\varepsilon}^{\mathsf{T}} \boldsymbol{w}^{\star m}$. The feedback between students and gates enters here in two terms, as can be seen by expressing $\boldsymbol{w}^{p\mathsf{T}} \boldsymbol{w}^{\star m} = ||\boldsymbol{w}^p|| \, ||\boldsymbol{w}^{\star m}|| \, \cos\left(\angle(\boldsymbol{w}^p, \boldsymbol{w}^{\star m})\right)$, where $\angle$ denotes the angle between two vectors. As observed in Fig. 2, both the *alignment* between students and teachers $\cos\left(\angle(\boldsymbol{w}^p, \boldsymbol{w}^{\star m})\right)$ and the *magnitude* of the students $||\boldsymbol{w}^p||$ control the gate switching speed.

As the vectors $\boldsymbol{w}^p$ are formed from the students' singular values, they scale proportionally to the bare matrix entries W_{ij}^p for random initialization (Marcenko-Pastur distribution). Early in learning, the small initialization will therefore attenuate gate changes by prolonging their effective timescale $\tau_c/||\boldsymbol{w}^p||$ (or equivalently, lower their learning rate).

As we demonstrate in Fig. 4D, these effects apply to both activation and inactivation of the gates, depending on the direction of the current error $\boldsymbol{\varepsilon}^\intercal \boldsymbol{w}^{\star m} \gtrless 0$.

The regularization in the system introduces a feedback loop between c^1 and c^2. In practice, the system quickly reaches a regime where both gates c^p are positive. In this case, the regularization term using the L^1-norm becomes $\nabla_{c^p}\mathcal{L}_{\text{reg}} \propto \sum_{p'} c^{p'} - 1$, reaching a minimum along the line $\sum_{p'} c^{p'} = 1$. In order to minimize the regularization loss, the upscaling of one gate c^p past 0.5 will result in the downscaling of the other gate $c^{p'}$, and vice versa. We note that this regularization term does not favor specialization by itself since the network can also attain zero loss in the unspecified forgetful solution with, for instance, $c^1 = c^2 = 0.5$.

The above dynamics mean that differences in student alignment separate the gates, as described by

$$\tau_c \frac{\mathrm{d}\bar{c}}{\mathrm{d}t} = \bar{w}_1\, \varepsilon_1 - \bar{w}_2\, \varepsilon_2 \tag{6}$$

We therefore see that the differences in gate activation are driven by the difference in specialization in the two components $\bar{w}_1$ and $\bar{w}_2$ and corresponding error components ε_1 and ε_2. Since the error components are of opposite sign following a context switch, $\frac{\mathrm{d}\bar{c}}{\mathrm{d}t}$ is maximized when the students are maximally specialized.

5.2.2 Flexible gates protect learned specialization

We now study the influence of c^p on $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{w}^p$. The gates allow for a switching mechanism that does not require a change in the weights. When continuing gradient descent on all parameters however, Eq. (1) will also entail a finite update to the wrong student.

If we Taylor-expand the update to second order, this update reads

$$\tau_w \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{w}^p \;\simeq\; c^p \boldsymbol{\varepsilon} \;+\; \frac{1}{2}\left(\left(\frac{\mathrm{d}}{\mathrm{d}t}c^p\right)\boldsymbol{\varepsilon} + c^p\left(\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\varepsilon}\right)\right)\, dt. \tag{7}$$

The first summand of the second term reflects the protection that arises from changes in gating $\frac{\mathrm{d}}{\mathrm{d}t}c^p = \boldsymbol{w}^{p\intercal}\boldsymbol{\varepsilon}$: a task switch to $\boldsymbol{y}^{\star m} = (0,1)^\intercal$ incurs an error $\boldsymbol{\varepsilon} \propto (-1,1)^\intercal$. For a specialized, but now incorrect student $\boldsymbol{w}^p \propto (1,\,0)^\intercal$, this term becomes $\frac{\mathrm{d}}{\mathrm{d}t}c^p = \boldsymbol{w}^{p\intercal}\boldsymbol{\varepsilon} < 0$ for the incorrect student. Together with the decreasing error in the last term $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\varepsilon}$, this reduces the student update from the leading-order first term $c^p\boldsymbol{\varepsilon}$. Importantly, this protection effect grows over training as the student's contribution to the error $\boldsymbol{w}^{p\intercal}\boldsymbol{\varepsilon}$ increases.

Alongside protection, flexible gates also accelerate specialization, as can be seen by considering $\boldsymbol{w}$ in specialization space,

$$\tau_w \frac{\mathrm{d}\bar{w}}{\mathrm{d}t} = \frac{1}{2}\bar{c}\,(\varepsilon_1 - \varepsilon_2) \tag{8}$$

This equation shows that the students specialize through two factors: the difference in error between the two components $\varepsilon_1 - \varepsilon_2$, and the difference in gate activation $\bar{c} = c^1 - c^2$.

5.3 Exact solutions to the learning dynamics describe protection and adaptation under symmetry in the flexible regime

In this section, we study exact solutions to the learning dynamics in Eq. (6) and Eq. (8) to describe the behavior of the model as it switches between tasks when it is already in the flexible regime. To solve the differential equations, we require the condition of symmetry where $\bar{w} = \bar{w}_1 = \bar{w}_2$.This condition is approximately true for specialized states in the flexible regime when (see Fig. A.10), and its persistence is theoretically guaranteed in the limit of strong L^1 regularization.

We use the method presented in Shi et al. [2022] to solve the resulting dynamics of the ratio between the expressions for $\frac{\mathrm{d}\bar{c}}{\mathrm{d}t}$ and $\frac{\mathrm{d}\bar{w}}{\mathrm{d}t}$

$$\frac{\tau_c}{\tau_w}\frac{\mathrm{d}\bar{c}}{\mathrm{d}\bar{w}} = 2\,\frac{\bar{w}\,(\varepsilon_1-\varepsilon_2)}{\bar{c}\,(\varepsilon_1-\varepsilon_2)} \tag{9}$$

which is a separable differential equation which can be solved up to an integration constant (see Appendix A.7.2). Plugging in initial conditions that correspond to complete specialization in the flexible regime $\bar{c}(0) = \bar{w}(0) = 1$, we obtain the exact dynamics of $\bar{w}$ as a function of $\bar{c}$ over the course of a block,

$$\bar{w} = \sqrt{1 - \frac{1}{2}\frac{\tau_c}{\tau_w}\,(1-\bar{c}^2)}. \tag{10}$$

This analytical solution accurately describes adaptation in the flexible regime (Fig. 4E). The relationship highlights the role of a shorter gate timescale τ_c in protecting the student's knowledge. Learning that comes from both students specializing towards the current teacher occurs outside of this subspace and becomes more important for low τ_c (see Appendix A.7.3).

6 Quantifying the range of the flexible regime across block length, regularization strength, and gate speed

To assess the roles of block length, regularization, and fast gate timescale (inverse gate learning rate) in establishing the flexible regime, we run two grid searches over the gate learning rate/regularization strength and block length each task is trained on, keeping the total time trained constant (such that models trained on shorter block lengths are trained over more block switches but equal amounts of data). For each set of hyperparameters we compute the total alignment (cosine similarity) between the entire concatenated set of teachers and students as a single overall measure of specialization in the network weights at the end of learning. We identify the boundaries of the flexible regime where specialization emerges in our model, dependent on block length, gate timescale, and regularization strength (Fig. 5). A priori, the block length dependence is surprising, as one might expect additional time spent in a block to be reversed by the equally-long subsequent block. However, we show in Appendix A.6 that gating breaks this time-reversal symmetry, and specialization grows with block length τ_B for fixed overall learning time t.

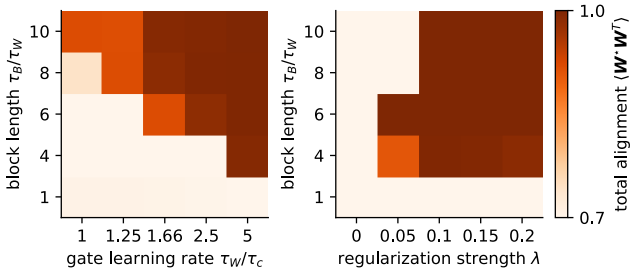

Figure 5: Model specialization emerges as a function of block length, gate learning rate, and regularization strength. The colorbar indicates total alignment (cosine similarity) between all sets of students and teachers considered collectively.

7 Inducing the flexible regime in a deep fully-connected neural network

Our NTA model uses a low-dimensional gating layer that gates computations from student networks. We sought to understand the necessity and role of this structure and to demonstrate a more general form of the model in a deep linear network with no architectural assumptions. Based on the analysis and results so far (Fig. 4D,5), we impose regularization and faster learning rates on the second layer of a 2-layer fully-connected network. Behaviorally, this network shows the signatures of the flexible regime with adaptation accelerating with each task switch experienced (Fig. A.4A).

To quantify specialization and gating behavior, we compute the cosine similarity between each row of the first hidden layer and the teachers and use this to sort the network into two students that align to the teachers they match best. We also permute the second layer columns to match the sorting of the first layer. We then visualize specialization of the sorted first hidden layer using the same measures as in the original NTA model. We also take the mean of each sorted student's second hidden layer to be its corresponding gate. Using this analysis, we find emergent gating in the second layer (Fig. A.4B) and specialization in the first (Fig. A.4C). Adaptation to later task switches takes place primarily in the second layer (Fig. A.4E).

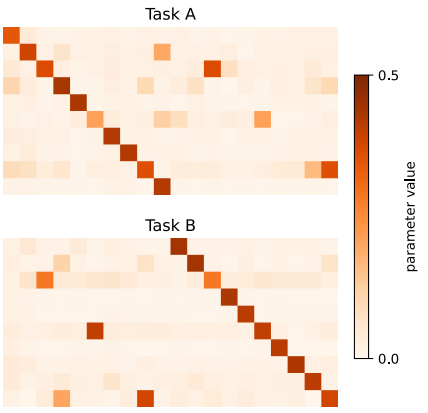

Figure 6: Task-specialized gating emerges in the second layer of a 2-layer network with faster second-layer learning rate and regularization. The sorted second layer weights at the last timestep of two different task blocks (one seed).

By visualizing the second layer of the sorted fully-connected network at the last timestep of two different task blocks, we indeed observe distinct gating behavior along the diagonal, specialized for each task (Fig. 6 for one seed). We compare this to the same fully-connected network trained without regularization which remains in the forgetful learning regime. We include visualizations of the unsorted second hidden layer for fully-connected networks arriving at both the gating and non-gating solutions for a single seed (Fig. A.5), as well as the sorted second hidden layer averaged across ten seeds (Fig. A.6) as supplement. Appendix A.4.2 discusses the potential for multiplicative gates to emerge in fully-connected architectures.

8 Flexible remapping of representations in nonlinear networks in two MNIST tasks

We next study whether NTA also works in larger, nonlinear systems. As a proof of concept, we investigate whether NTA can help a neural network switch between two nonlinearly-transformed versions of the MNIST dataset [Deng, 2012]. The first task is the conventional MNIST task. The second is a permuted version of MNIST where the image of a digit is sorted based on its parity according to the function $y \rightarrow \lfloor y/2 \rfloor + 5 \times (y\%2)$, where $\%$ is the modulo operation (see Fig. 7A). We

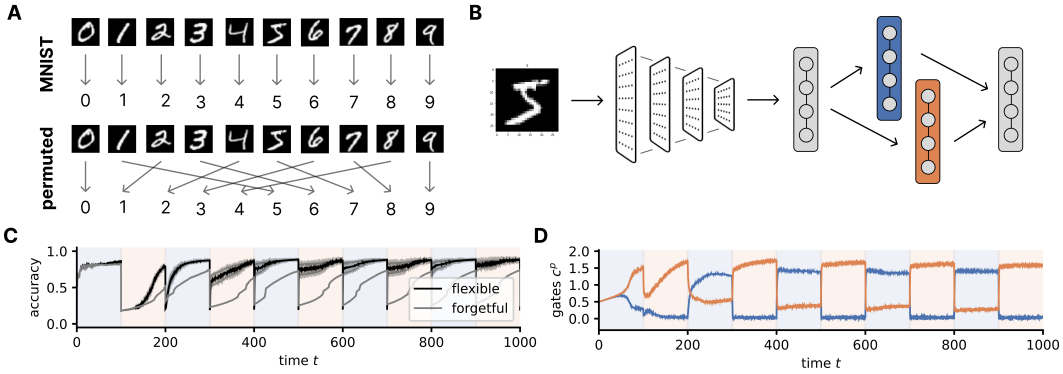

Figure 7: Learning flexible neural task abstractions in a nonlinear character recognition setting. **A.** We formulate two tasks, the original and a permuted version of MNIST. **B.** We embed the NTA system into a larger pretrained convolutional neural network architecture. **C.** Accuracy reached on the MNIST test set as a function of time for both (*black*) the NTA network and (*gray*) the original CNN. The two tasks are presented sequentially in blocks for both (*blue shading*) MNIST and (*orange shading*) the permuted version. **D.** The activation of the two gating units as a function of time. We show mean and standard error with 10 seeds.

pre-train a convolutional neural network (CNN) on MNIST to learn useful representations, achieving about 90% accuracy on the test set. We then train an NTA system beginning from the final hidden layer representations that feeds into the same sigmoid nonlinearity (see Fig. 7B). We again induce the flexible regime using regularization and fast timescales, and contrast performance with a forgetful model (see Appendix B.7). We find that the flexible model learns to recover its original accuracy quickly after the first task switches whereas the forgetful one needs to continuously re-learn the task, as evaluated on the MNIST test set (Fig. 7C). The activity in the gating units reflects selective activity (Fig. 7D). To further test the range of NTA, we examine how much these results depend on orthogonality of the task space by formulating two tasks based on real-world groupings of clothing in fashionMNIST [Xiao et al., 2017] that have different amounts of shared structure. We find that rapid task switching occurs in both settings at a similar speed (Fig. A.15).

9 Relations to multi-task learning in humans

Our model captures several aspects of human learning. Humans update task knowledge in proportion to how well each probable task explains current experience [Castañón et al., 2021]. Analogously in our model, weight updates are gated with the corresponding gating variable, which in turn is active proportional to how well the weights behind it capture the target computation.

Humans show faster task switching with more practice on the tasks involved. In our model, we saw that the gates change faster as weights specialize to the tasks, which facilitated faster adaptation after block switches. NTA shows a qualitative fit (Fig. 8) to humans trained on alternating tasks [Steyvers et al., 2019]. In contrast, a forgetful model shows a deceleration, possibly due to being far from optimal initialization [Dohare et al., 2024] after task switches.

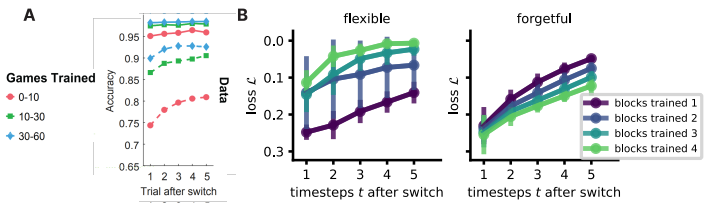

Figure 8: Comparing performance after a task switch in humans and NTA model. A. Steyvers et al. [2019] report performance of humans learning two alternating tasks (CC BY-NC-ND 4.0 license). **B.** After a block switch, loss comparison between the flexible (*left*) and the forgetful (*right*) NTA model shows opposite trends with further training on switching speed. Bars are standard error with 10 seeds.

10 Conclusions and future work

Our work proposes a parsimonious addition to neural network models where gradient descent in neural activity space induces abstract task representations allowing for behavioral flexibility and compositional generalization. Such a model that discovers tasks as a context for learning might map better to human cognition, and we expect future work to make further connections. We investigate nonnegativity and normalization as relevant mechanisms for specialization: these regularization constraints map conceptually to neuronal electrophysiological properties, which we interpret as support for the relevance of gradient descent in neural space. In this study, we focus on simple two-layer networks, but in principle the framework is applicable to other architectures such as recurrent networks or Transformer architectures. While we provide a proof of concept of a non-linear image classifier, we see future work providing more real-world applications of the framework.

Author contributions

All authors contributed to manuscript writing and conceptualization of the work. AP conceptualized and implemented the experiments, and contributed to the idea, the theory, and the model simulator. JB conceptualized and developed the theory, developed the model simulator, and contributed to the idea and experiments. KS conceptualized and developed the theory, conceived of the idea and the link to cognitive flexibility, and contributed to experiments. AH directed the project, contributed to the idea and the experiments, and provided feedback on the theory.

Acknowledgements

We thank Stefano Sarao Mannelli and Pedro A.M. Mediano for thoughtful discussions. We would also like to acknowledge and thank the organizers of the Analytical Connectionism Summer School, at which AP, JB, and KS first met. AH is funded by Collaborative Research in Computational Neuroscience award (R01-MH132172). AP is funded by the Imperial College London President's PhD Scholarship. KS is funded by a Cusanuswerk Doctoral Fellowship. JB is supported by the Gatsby Charitable Foundation (GAT3850). This work was supported by a Sir Henry Dale Fellowship from the Wellcome Trust and Royal Society (216386/Z/19/Z) to AS, and the Sainsbury Wellcome Centre Core Grant from Wellcome (219627/Z/19/Z) and the Gatsby Charitable Foundation (GAT3755).

References

Samuel J Gershman and Yael Niv. Learning latent structure: carving nature at its joints. *Current Opinion in Neurobiology*, 20(2):251–256, April 2010. ISSN 0959-4388. doi: 10.1016/j.conb.2010.02.008. URL https://www.sciencedirect.com/science/article/pii/S0959438810000309.

Linda Q. Yu, Robert C. Wilson, and Matthew R. Nassar. Adaptive learning is structure learning in time. *Neuroscience & Biobehavioral Reviews*, 128:270–281, September 2021. ISSN 0149-7634. doi: 10.1016/j.neubiorev.2021.06.024. URL https://www.sciencedirect.com/science/article/pii/S0149763421002657.

Santiago Herce Castañón, Pedro Cardoso-Leite, Irene Altarelli, C. Shawn Green, Paul Schrater, and Daphne Bavelier. A mixture of generative models strategy helps humans generalize across tasks. *bioRxiv*, page 2021.02.16.431506, January 2021. doi: 10.1101/2021.02.16.431506. URL http://biorxiv.org/content/early/2021/02/16/2021.02.16.431506.abstract.

Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in hippocampus and prefrontal cortex. *Cell*, 183(4):954–967.e21, November 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.09.031. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8451959/.

Sina Tafazoli, Flora M. Bouchacourt, Adel Ardalan, Nikola T. Markov, Motoaki Uchimura, Marcelo G. Mattar, Nathaniel D. Daw, and Timothy J. Buschman. Building compositional tasks with shared neural subspaces. *bioRxiv*, 2024. doi: 10.1101/2024.01.31.578263. URL https://www.biorxiv.org/content/early/2024/03/22/2024.01.31.578263. Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2024/03/22/2024.01.31.578263.full.pdf.

Timo Flesch, Jan Balaguer, Ronald Dekker, Hamed Nili, and Christopher Summerfield. Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, 115(44):E10313–E10322, October 2018. doi: 10.1073/pnas.1800755115. URL https://www.pnas.org/doi/full/10.1073/pnas.1800755115. Publisher: Proceedings of the National Academy of Sciences.

Andre O. Beukers, Silvy H. P. Collin, Ross P. Kempner, Nicholas T. Franklin, Samuel J. Gershman, and Kenneth A. Norman. Blocked training facilitates learning of multiple schemas. *Communications Psychology*, 2(1):1–17, April 2024. ISSN 2731-9121. doi: 10.1038/s44271-024-00079-4. URL https://www.nature.com/articles/s44271-024-00079-4. Publisher: Nature Publishing Group.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, January 1989. doi: 10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing Change: Continual Learning in Deep Neural Networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, December 2020. ISSN 1364-6613. doi: 10.1016/j.tics.2020.09.004. URL https://www.sciencedirect.com/science/article/pii/S1364661320302199.

Ali Hummos, Felipe del Río, Brabeeba Mien Wang, Julio Hurtado, Cristian B. Calderon, and Guangyu Robert Yang. Gradient-based inference of abstract task representations for generalization in neural networks, 2024. URL https://arxiv.org/abs/2407.17356. _eprint: 2407.17356.

Ali Hummos. Thalamus: a brain-inspired algorithm for biologically-plausible continual learning and disentangled representations. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6orC5MvgPBK.

Martin V. Butz, David Bilkey, Dania Humaidan, Alistair Knott, and Sebastian Otte. Learning, planning, and control in a monolithic neural event inference architecture. *Neural Networks*, 117:135–144, September 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.05.001. URL https://www.sciencedirect.com/science/article/pii/S0893608019301339.

Qihong Lu, Ali Hummos, and Kenneth A Norman. Episodic memory supports the acquisition of structured task representations. *bioRxiv*, page 2024.05.06.592749, January 2024. doi: 10.1101/2024.05.06.592749. URL http://biorxiv.org/content/early/2024/05/07/2024.05.06.592749.abstract.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wolczyk, Alexandra Maria Proca, Johannes Von Oswald, Razvan Pascanu, Joao Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H98CVcX1eh.

Mark Steyvers, Guy E. Hawkins, Frini Karayanidis, and Scott D. Brown. A large-scale analysis of task switching practice effects across the lifespan. *Proceedings of the National Academy of Sciences*, 116(36):17735–17740, September 2019. doi: 10.1073/pnas.1906788116. URL https://www.pnas.org/doi/full/10.1073/pnas.1906788116. Publisher: Proceedings of the National Academy of Sciences.

E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24:167–202, 2001. ISSN 0147-006X. doi: 10.1146/annurev.neuro.24.1.167.

Tobias Egner. Principles of cognitive control over task focus and task switching. *Nature Reviews Psychology*, 2(11):702–714, November 2023. ISSN 2731-0574. doi: 10.1038/s44159-023-00234-4. URL https://www.nature.com/articles/s44159-023-00234-4. Publisher: Nature Publishing Group.

Kai Sandbrink and Christopher Summerfield. Modelling cognitive flexibility with deep neural networks. *Current Opinion in Behavioral Sciences*, 57:101361, June 2024. ISSN 2352-1546. doi: 10.1016/j.cobeha.2024.101361. URL https://www.sciencedirect.com/science/article/pii/S2352154624000123.

Sebastian Musslick and Jonathan D. Cohen. Rationalizing constraints on the capacity for cognitive control. *Trends in Cognitive Sciences*, 25(9):757–775, September 2021. ISSN 1879-307X. doi: 10.1016/j.tics.2021.06.001.

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, February 2019. ISSN 1546-1726. doi: 10.1038/s41593-018-0310-2. URL https://www.nature.com/articles/s41593-018-0310-2.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/10.1073/pnas.1611835114. Publisher: Proceedings of the National Academy of Sciences.

Nicolas Y. Masse, Gregory D. Grant, and David J. Freedman. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, 115(44):E10467–E10475, October 2018. doi: 10.1073/pnas.1803839115. URL https://www.pnas.org/doi/10.1073/pnas.1803839115. Publisher: Proceedings of the National Academy of Sciences.

Dongkai Wang and Shiliang Zhang. Contextual Instance Decoupling for Robust Multi-Person Pose Estimation. pages 11060–11068, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Contextual_Instance_Decoupling_for_Robust_Multi-Person_Pose_Estimation_CVPR_2022_paper.html.

Laura N. Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, July 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01668-6. URL https://www.nature.com/articles/s41593-024-01668-6. Publisher: Nature Publishing Group.

Nicolas Y. Masse, Matthew C. Rosen, Doris Y. Tsao, and David J. Freedman. Flexible cognition in rigid reservoir networks modulated by behavioral context, May 2022. URL https://www.biorxiv.org/content/10.1101/2022.05.09.491102v2. Pages: 2022.05.09.491102 Section: New Results.

Robert A. Jacobs, Michael I. Jordan, and Andrew G. Barto. Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks. *Cognitive Science*, 15(2):219–250, 1991. doi: https://doi.org/10.1207/s15516709cog1502_2. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1502_2. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1502_2.

Michael I. Jordan and Robert A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6(2):181–214, March 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.2.181. URL https://doi.org/10.1162/neco.1994.6.2.181. _eprint: https://direct.mit.edu/neco/article-pdf/6/2/181/812708/neco.1994.6.2.181.pdf.

Ben Tsuda, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117(47):29872–29882, November 2020. doi: 10.1073/pnas.2009591117. URL https://www.pnas.org/doi/abs/10.1073/pnas.2009591117. Publisher: Proceedings of the National Academy of Sciences.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

Louis Kirsch, Julius Kunze, and D. Barber. Modular Networks: Learning to Decompose Neural Computation. November 2018. URL https://www.semanticscholar.org/paper/Modular-Networks%3A-Learning-to-Decompose-Neural-Kirsch-Kunze/0b50b9e103e19d87c2d30ed0d157d8379320ce6f.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and B. Schölkopf. Recurrent Independent Mechanisms. *ArXiv*, 2019.

John Geweke. Interpretation and inference in mixture models: Simple MCMC works. *Computational Statistics & Data Analysis*, 51(7):3529–3550, April 2007. ISSN 0167-9473. doi: 10.1016/j.csda.2006.11.026. URL https://www.sciencedirect.com/science/article/pii/S0167947306004506.

Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to Combine Top-Down and Bottom-Up Signals in Recurrent Neural Networks with Attention over Modules. *arXiv:2006.16981 [cs, stat]*, November 2020. URL http://arxiv.org/abs/2006.16981. arXiv: 2006.16981.

Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. Improving Expert Specialization in Mixture of Experts. *arXiv preprint arXiv:2302.14703*, 2023.

Martin Barry and Wulfram Gerstner. Fast Adaptation to Rule Switching using Neuronal Surprise, September 2022. URL https://www.biorxiv.org/content/10.1101/2022.09.13.507727v1. Pages: 2022.09.13.507727 Section: New Results.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. tex.creationdate: 2022-07-07T18:40:31 tex.modificationdate: 2022-07-07T18:40:39.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23): 11537–11546, June 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/10.1073/pnas.1820226116. Publisher: Proceedings of the National Academy of Sciences.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The Neural Race Reduction: Dynamics of Abstraction in Gated Networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19287–19309. PMLR, July 2022. URL https://proceedings.mlr.press/v162/saxe22a.html.

Jianghong Shi, Eric Shea-Brown, and Michael A. Buice. Learning dynamics of deep linear networks with multiple pathways. *Advances in Neural Information Processing Systems*, 35:34064–34076, December 2022. ISSN 1049-5258.

Jin Hwa Lee, Stefano Sarao Mannelli, and Andrew Saxe. Why Do Animals Need Shaping? A Theory of Task Composition and Curriculum Learning. *arXiv preprint arXiv:2402.18361*, 2024.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Publisher: IEEE.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, August 2017. arXiv: cs.LG/1708.07747 [cs.LG].

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632 (8026):768–774, August 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07711-7. URL https://www.nature.com/articles/s41586-024-07711-7. Publisher: Nature Publishing Group.

Alexander Atanasov, Blake Bordelon, and C. Pehlevan. Neural Networks as Kernel Learners: The Silent Alignment Effect. *ArXiv*, October 2021. URL https://www.semanticscholar.org/paper/Neural-Networks-as-Kernel-Learners%3A-The-Silent-Atanasov-Bordelon/ccd3631a4509aac2d71c320a6ac677f311d94b05.

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Advances in Neural Information Processing Systems*, 35:6615–6629, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2b3bb2c95195130977a51b3bb251c40a-Abstract-Conference.html.

Alexander Atanasov, Blake Bordelon, Sabarish Sainathan, and Cengiz Pehlevan. The onset of variance-limited behavior for networks in the lazy and rich regimes. December 2022. arXiv: 2212.12147 [stat.ML] tex.creationdate: 2022-12-28T08:24:33 tex.modificationdate: 2022-12-28T08:25:21.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, February 2020. URL http://arxiv.org/abs/1806.07572. arXiv: 1806.07572.

Appendix

Overview

We structure the Appendix as follows:

We first provide additional material on our results in Appendix A. Appendix A.1 derives the reduction to the 2D equivalent model. Appendix A.2 shows that even without a differential timescale between gates and weights, high-rank students will learn more slowly compared to gates. Appendix A.3 shows that networks with more paths than teachers will split their representations across paths unless a cost is associated with representation. Appendix A.4 provides simulations and derivations on how gating behavior emerges in an architecture without explicit pathways but with two layers, where one layer emergently takes on the role of gates, and the second layer becomes compartmentalized. Appendix A.4 contains additional results for the fully-connected network. Appendix A.5 shows how the specialized representation incentivized by the virtuous cycle discussed in the main text leads to a faster reduction in loss compared to an unspecialized solution. Appendix A.6 provides an approximate theoretical explanation for the beneficial effect of long blocks towards specialization through symmetry breaking in an effective potential. Appendix A.7 provides approximate closed-form solutions when operating in the flexible regime. Appendix A.8 provides detail on how the model generalizes to new tasks by leveraging existing abstractions. Appendix A.9 shows that the flexible regime largely persists and slowly decays when the orthogonality assumption between teachers is relaxed. Appendix A.10 shows that the model can adapt in a few-shot fashion after a block switch, extending the results from the main text where gradients are calculated on many samples.

We then provide additional technical details in Appendix B. In Appendix B.1, we provide a notation table. Appendix B.2 lists parameters used for simulations. Appendix B.3 discusses the regularization that we use, in particular why it does not incentivize a flexible over a forgetful solution. Appendix B.4 details on how we calculate alignments between teachers and students, both for the per-student and per-neuron gating models. Appendix B.5 discusses how we choose model parameters.

A Additional details on main text

A.1 Derivation of reduction to 2D equivalent model

We will here show that the dynamics of the model can be reduced to an effective model that acts in a 2D space spanned by the singular teacher vectors across both tasks m.

Recall the task loss as the mean-squared error

$$\begin{aligned}\mathcal{L}_{\text{task}} &= \frac{1}{2Bd_{\text{out}}}\sum_{b}^{B}\|\boldsymbol{y}_b^{\star m}-\boldsymbol{y}_b\|^2 \\ &= \frac{1}{2Bd_{\text{out}}}\sum_{b}^{B}\|\boldsymbol{W}^{\star m}\boldsymbol{x}_b-\sum_p c^p\boldsymbol{W}^p\boldsymbol{x}_b\|^2\end{aligned}$$

for a batch $\boldsymbol{X}=(\boldsymbol{x}_b)_{b=1\dots B}$ of size B.

Following the approach in Saxe et al. [2013], we assume the input data is whitened, such that the batch average $\frac{1}{B}\boldsymbol{X}\boldsymbol{X}^\intercal\approx\boldsymbol{I}_{d_{\text{in}}}$, and the learning rate τ^{-1} is small (i.e., the *gradient flow* regime). Then, the batch gradient reads as a differential equation that simplifies as

$$\tau_w\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{W}^p = \frac{1}{Bd_{\text{out}}}c^p\left(\boldsymbol{W}^{\star m}\boldsymbol{X}-\textstyle\sum_{p'}c^{p'}\boldsymbol{W}^{p'}\boldsymbol{X}\right)\boldsymbol{X}^\intercal \tag{11}$$

$$\approx c^p\left(\boldsymbol{W}^{\star m}-\textstyle\sum_{p'}c^{p'}\boldsymbol{W}^{p'}\right). \tag{12}$$

For each teacher m and singular value decomposition along a mode α ($\boldsymbol{u}_\alpha^{\star m}, s_\alpha^{\star m}, \boldsymbol{v}_\alpha^{\star m}$), we can project this equation to get

$$\tau_w \frac{\mathrm{d}}{\mathrm{d}t}\Big(\underbrace{\boldsymbol{u}_\alpha^{\star m\intercal}\boldsymbol{W}^p\boldsymbol{v}_\alpha^{\star m}}_{s^p_{m,\alpha}}\Big) = c^p\left(\boldsymbol{u}_\alpha^{\star m\intercal}\boldsymbol{W}^{\star m}\boldsymbol{v}_\alpha^{\star m} - \boldsymbol{u}_\alpha^{\star m\intercal}\textstyle\sum_{p'} c^{p'}\boldsymbol{W}^{p'}\boldsymbol{v}_\alpha^{\star m}\right) \tag{13}$$

$$= c^p\left(\quad s_\alpha^{\star m} \quad - \quad \textstyle\sum_{p'} c^{p'} s^{p'}_{m,\alpha} \quad\right).$$

The student singular vectors $\boldsymbol{u}_\alpha^p, \boldsymbol{v}_\alpha^p$ have been shown to align to those of the current teacher $\boldsymbol{u}_\alpha^{\star m}, \boldsymbol{v}_\alpha^{\star m}$ early in learning [Atanasov et al., 2021]. After training on both teachers, the student can therefore be fully described in terms of the coefficients $\left\{s^p_{m,\alpha}\right\}_{m\alpha}$ in the basis spanned by the α-singular vectors of both teachers, decoupling from the other singular value dimensions. If all singular vectors across two teachers m are pairwise orthogonal, these projections form an orthogonal basis. The components outside of this projection will have finite error in all context and therefore exponentially decay to 0 [Braun et al., 2022].

This reduction allows us to study learning in a simpler and more interpretable model. For the case where $M = P = 2$ which we consider here for simplicity, we can therefore reinterpret each model α-component as vectors $\boldsymbol{w}^1 \equiv (s^{p=1}_{m=1,\alpha},\, s^{p=1}_{m=2,\alpha})^\intercal$, $\boldsymbol{w}^2 \equiv (s^{p=2}_{m=1,\alpha},\, s^{p=2}_{m=2,\alpha})^\intercal$ in $\mathbb{R}^2$:

$$\begin{aligned}\boldsymbol{y} &= c^1\boldsymbol{w}^1 + c^2\boldsymbol{w}^2 \\ &= c^1\begin{pmatrix} s^1_{1,\alpha} \\ s^1_{2,\alpha}\end{pmatrix} + c^2\begin{pmatrix} s^2_{1,\alpha} \\ s^2_{2,\alpha}\end{pmatrix}\end{aligned} \tag{14}$$

and redefine the context-dependent target vector $\boldsymbol{y}^{\star m}$ accordingly.

The reduced model follows the update equations

$$\tau_w \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{w}^p = c^p\left(\boldsymbol{y}^{\star m} - \boldsymbol{y}\right) \tag{15}$$

$$\tau_c \frac{\mathrm{d}}{\mathrm{d}t}c^p = \boldsymbol{w}^{p\intercal}\left(\boldsymbol{y}^{\star m} - \boldsymbol{y}\right). \tag{16}$$

Notably, both updates depend on the full error term $\boldsymbol{\varepsilon} \coloneqq (\boldsymbol{y}^{\star m} - \boldsymbol{y})$ with both paths entering into $\boldsymbol{y}$. The first equation moves the student in the direction of the current total misestimation of the active teacher $\boldsymbol{\varepsilon}$. The second equation changes the gating of the current path according to the alignment of the path $\boldsymbol{w}^p$ to the current vectorial error, reflecting the contribution of the path to the mismatch.

In Fig. A.1, we simulate the models side by side and show that the reduced model matches the dynamics of the full model.

A.1.1 Reduction in terms of teacher row vectors

In the main text and the previous section, we consider a reduction that follows from projecting onto the eigenspace of the matrices. However, a similar reduction is possible by considering each row β independently, and considering the row vectors of the two teachers $\left(\boldsymbol{w}_\beta^{\star m}\right)_m$ as a basis for that row. Like in the projection in terms of the eigenspace, the out-of-projection component of the students decays exponentially. We can then consider a single row of the teacher-student system to function as a mode α above, with a row of the student path p becoming $\boldsymbol{w}^p = w_1^p\,\boldsymbol{w}^{\star 1} + w_2^p\,\boldsymbol{w}^{\star 2}$ so that we can write

$$\begin{aligned}\boldsymbol{y} &= c^1\boldsymbol{w}^1 + c^2\boldsymbol{w}^2 \\ &= c^1\begin{pmatrix} w^1_{1,\beta} \\ w^1_{2,\beta}\end{pmatrix} + c^2\begin{pmatrix} w^2_{1,\beta} \\ w^2_{2,\beta}\end{pmatrix}\end{aligned} \tag{17}$$

where we aggregate over rows β. This formulation only requires pairwise orthogonality between rows $\boldsymbol{w}_i^{\star 1} \cdot \boldsymbol{w}_i^{\star 2} = 0$ to fully decouple the dynamics of the system, but does not extend as elegantly to considering deeper students or low-rank solutions.

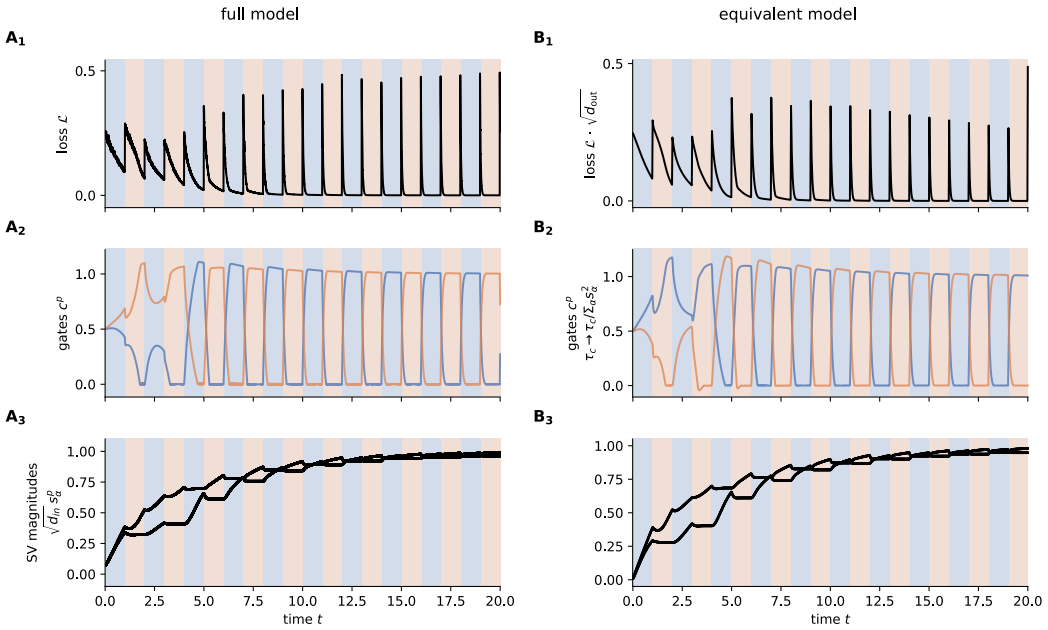

Figure A.1: Simulation of dynamics of full and reduced model. The equivalent reduced model effectively captures the dynamics of the full model in terms of loss ($\mathbf{A}_1\mathbf{.}$, $\mathbf{B}_1\mathbf{.}$), gates ($\mathbf{A}_2\mathbf{.}$, $\mathbf{B}_2\mathbf{.}$), and singular value magnitude ($\mathbf{A}_3\mathbf{.}$, $\mathbf{B}_3\mathbf{.}$).

A.2 High-dimensional students learn slower

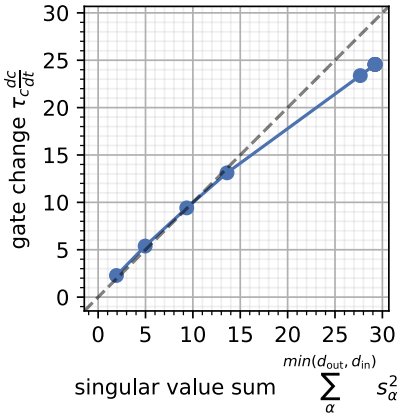

Figure A.2: High-dimensional students learn slower. Gate change $\tau_c \frac{\mathrm{d}}{\mathrm{d}t} c$ as a function of teacher dimensionality/rank (i.e., non-zero singular values). Weight scaling is chosen such that input and output components take unit scale, $y_i = \mathcal{O}(1)$, $x_j = \mathcal{O}(1)$.

An intuition one might have for the model dynamics is that the weight matrices comprise of more parameters and therefore may respond more slowly under gradient descent. Here, we discuss the formal conditions under which this indeed is the case.

For simplicity, we consider a one-path model $\boldsymbol{y} = c\boldsymbol{W}\boldsymbol{x}$, with $\boldsymbol{y} \in \mathbb{R}^{d_{\text{out}}}$, $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$. We now choose the scaling $y_i = \mathcal{O}(1)$, $x_j = \mathcal{O}(1)$ on input and output, which means that entries do not depend on the respective vector dimensionalities. This is a natural assumption, for example, if y_i are label indicators and x_j are pixel brightness values of an image. Recall the model loss $\mathcal{L}_{\text{task}} = {}^1\!/_2 \langle \boldsymbol{y}^{\star m} - \boldsymbol{y} \rangle^2$.

Then, the batch-averaged c-gradient reads

$$
\begin{aligned}
\tau_c \frac{\mathrm{d}}{\mathrm{d}t} c = -\nabla_c \mathcal{L}_{\text{task}} &= \text{Tr}\left[(\boldsymbol{y}^{\star m} - \boldsymbol{W}\boldsymbol{x})\boldsymbol{x}^\intercal \boldsymbol{W}^\intercal\right], \\
&= \text{Tr}\left[\boldsymbol{W}^\star \boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{W}^\intercal - \boldsymbol{W}\boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{W}^\intercal\right] \\
\langle \circ \rangle_B &\to \text{Tr}\left[\boldsymbol{W}^\star \boldsymbol{W}^\intercal - \boldsymbol{W}\boldsymbol{W}^\intercal\right] \\
&\approx \text{Tr}\left[\boldsymbol{U}\boldsymbol{S}^\star \boldsymbol{V}\boldsymbol{V}^\intercal \boldsymbol{S}\boldsymbol{U}^\intercal - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}\boldsymbol{V}^\intercal \boldsymbol{S}\boldsymbol{U}^\intercal\right] \\
&= \text{Tr}\left[\boldsymbol{S}^\star \boldsymbol{S} - \boldsymbol{S}^2\right] \\
&= \sum_{\alpha}^{\min(d_{\text{out}}, d_{\text{in}})} \left(s^\star_\alpha s_\alpha - s^2_\alpha\right).
\end{aligned}
$$

Here, we used the Gaussian i.i.d. initialization of $\boldsymbol{x}$ to take an expectation for large batch size ($\langle \boldsymbol{x}\boldsymbol{x}^\intercal \rangle_B = \boldsymbol{I}_{d_{\text{in}}}$), the SVD of $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\intercal$, orthonormalization of singular vectors $\mathbf{U}^\intercal\mathbf{U}$, $\mathbf{V}^\intercal\mathbf{V}$, and the cyclic property of the trace Tr. We also assumed in the fourth row that the student singular vectors have already undergone Silent Alignment [Atanasov et al., 2021] to match the teachers, as discussed in the main text.

From the last row, we observe that the updates to c tend to scale with the number of nonzero singular values, i.e. the rank of the teachers.

For the fan-in scaling $W_{ij} \sim \mathcal{N}(0,\, \sigma^2/d_{\text{in}})$ that is compatible with $x_i = \mathcal{O}(1)$, $y_i = \mathcal{O}(1)$, we have $s_\alpha = \mathcal{O}(\sigma)$ independent of dimensionality (Marcenko-Pastur distribution). If teacher and students are initialized according to this scaling, students will respond relatively slower compared to gates as their dimension $\min(d_{\text{in}}, d_{\text{out}})$ grows, as the student gradient (Eq. (11) or Eq. (13)) does not involve a sum that scales with dimensionality.

A.3 Representational cost in under-specified model

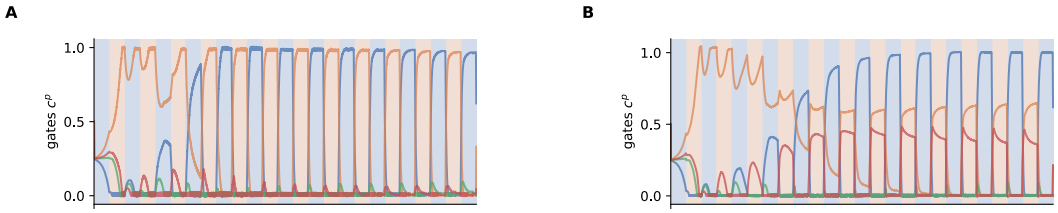

Figure A.3: Redundant paths become inactive when representation is costly. Gating variables like in Fig. 1B, but with more paths than teacher tasks ($P = 4 > M = 2$). **A.** Only under representational cost on the weights, students that are preferably aligned due to the random initialization specialize to the $M = 2$ teachers, whereas other gates decay to 0. **B.** Without representational cost, the model uses multiple paths for tasks and thus has multiple gates active at the same time for a single teacher.

In our initial Eq. (1, NTA), we have introduced a model in which the number of paths P of the architecture matches the number of available tasks. What happens if this match is not present? If the under-specified case $P < M$, the model's expressiveness hinders adaptation. It is however not clear what will happen in the over-specified case $P > M$. In absence of any regularization on the weights $\boldsymbol{W}^p$, the model will not devote only $P' = M < P$ paths to match the task. Rather, in accordance with the theory by Shi et al. [2022], the model will in general split its paths over the available tasks. This behavior is due to the absence of a "representational cost" of having multiple paths active at the same time. We find that this effect is reduced only when introducing an L^2-regularization $\frac{\lambda_W}{2PD_{\text{in}}} \sum_{ijp} (W^p_{ij})^2$, $\lambda_W = 0.77$. This term additionally penalizes weight magnitude leads to the decay of inactive paths. We show this behavior in Fig. A.3.

A.4 Gating-based solution emerges in a fully-connected network

As described in the main text, we induce the flexible gating regime in a fully-connected network by applying regularization and a faster learning rate to the second layer and compare to a forgetful

(unregularized) fully-connected network. Details of the sorting procedure used to identify and visualize this gating-based solution are described in Appendix B.4.1. We find that the flexible fully-connected network exhibits behavior that is qualitatively similar to the flexible NTA (Fig. A.4). By visualizing the second layer at the end of training on different tasks in the flexible regime, we observe that the network upweights single units in each row (Fig. A.5A, A.6A), which act as gates for the first layer rows. Instead, in the forgetful regime, the network has multiple upweighted units in each row and the units do not change behavior across different tasks, exhibiting a lack of task-specificity and gating-like behavior (Fig. A.5B, A.6B).

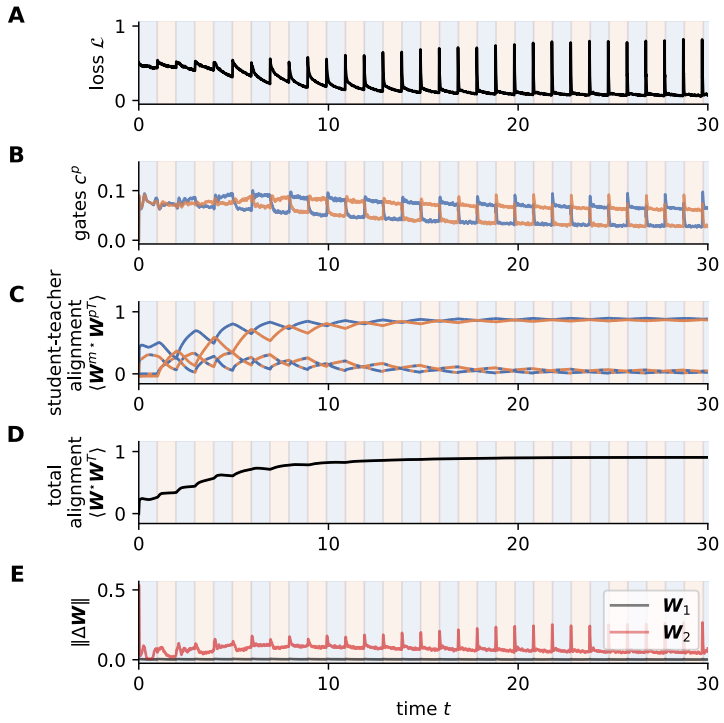

Figure A.4: Gating-based solution emerges in a fully-connected network with regularized second layer weights and a faster second layer learning rate. A. Loss during learning. **B.** The dynamics of the sorted gating variables. **C.** Alignment between the sorted students in the first layer and the teachers. **D.** Total alignment between the entire set of teachers and students. **E.** The norm of the gradient of the first (*black*) and second (*red*) hidden layer of the fully-connected network.

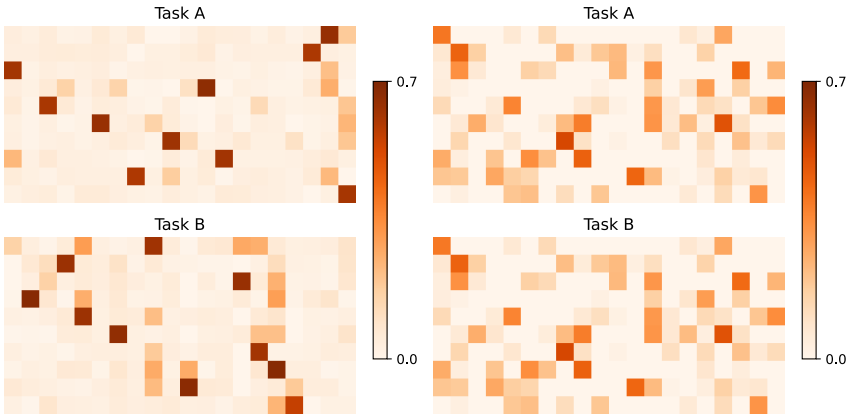

Figure A.5: Regularized, but not non-regularized, fully-connected network specializes single neurons in each row as 'gates' per task and exhibits specificity based on task. Visualization of the unsorted second hidden layer of the flexible (*left*) and forgetful (*right*) fully-connected network for a single seed.

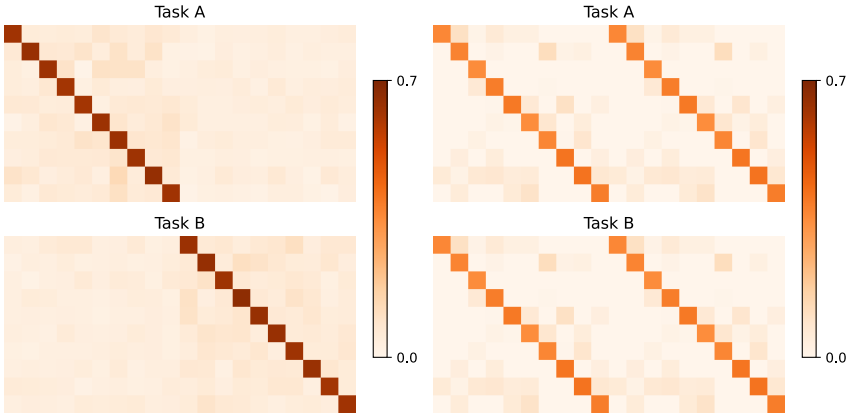

Figure A.6: Second hidden layer of regularized, but not non-regularized, fully-connected network exhibits clear task-specific gating across the diagonals of the matrix. Visualization of the sorted second hidden layer of the flexible (*left*) and forgetful (*right*) fully-connected network averaged over 10 seeds.

A.4.1 Model specialization as a function of block size, gate timescale, and regularization strength in fully-connected network

We perform two hyperparameter searches to illustrate the joint effects of block length, second layer learning rate, and regularization strength on the fully-connected network, similar to that we perform on the NTA model in the main text. We run the fully-connected network on each set of hyperparameters and report the total alignment of sorted teachers and students at the end of training as an overall measure of specialization, fixing all other hyperparameters (see Appendix B.5 for more details). We observe that the same components of block length, fast second layer learning rate, and regularization are important for specialization to emerge in the fully-connected network (Fig. A.7), just as in the NTA model.

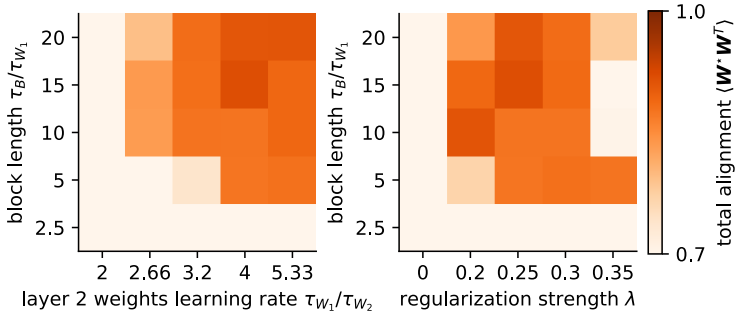

Figure A.7: Model specialization emerges as a function of block length, second hidden layer learning rate, and regularization strength in fully-connected network. The colorbar indicates total alignment (cosine similarity) between all sets of students and teachers considered collectively.

A.4.2 Possibility of emergence of gating in two-layer network

In this work, we have analyzed a linear architecture with an explicit architectural gating structure,

$$y_i = \sum_{p=1}^{P} \sum_{j=1}^{d_{\text{in}}} c^p W_{ij}^p x_j = (\boldsymbol{c} \odot \boldsymbol{W}\boldsymbol{x})_i, \tag{18}$$

where we have notationally stacked the students $\boldsymbol{W} \coloneqq (\boldsymbol{W}^p)_{p=1\ldots P}$ into a vector, such that $\boldsymbol{W} \in \mathbb{R}^{P \times d_{\text{out}} \times d_{\text{in}}}$, $\boldsymbol{c} \in \mathbb{R}^P$. $\odot$ here denotes the Hadamard (element-wise) product.

Prior work has considered deep linear networks [Saxe et al., 2019, Atanasov et al., 2022, Shi et al., 2022, Braun et al., 2022], which led us to study such fully-connected network in the main text,

$$y_i = \sum_{j=1}^{d_{\text{in}}} \sum_{h=1}^{d_{\text{hid}}} W_{ih}^{(2)} W_{hj}^{(1)} x_j = (\boldsymbol{W}^{(2)} \boldsymbol{W}^{(1)} \boldsymbol{x})_i. \tag{19}$$

The gated network considers gating as a multiplicative effect on each output unit i (or equivalently, input unit j), whereas the deep network invokes an additional all-to-all weighted summation. As such, Eq. (19) does not incorporate any modular structure, yet formally resembles Eq. (18). To further analyze how these settings connect, we decompose the tasks as $W_{ij}^{\star m} = \sum_\alpha U_{i\alpha}^{\star m} s_\alpha^{\star m} V_{\alpha j}^{\star m}$, and write the student layer matrices as SVDs $W_{ih}^{(2)} = \sum_\alpha U_{i\alpha}^{(2)} s_\alpha^{(2)} V_{\alpha h}^{(2)\intercal}$, $W_{hi}^{(1)} = \sum_\alpha U_{h\alpha}^{(1)} s_\alpha^{(1)} V_{\alpha j}^{(1)\intercal}$. The overall model Eq. (19) then reads

$$\begin{aligned} y_i &= \sum_{j}^{d_{\text{in}}} \sum_{\alpha}^{\min(d_{\text{out}}, d_{\text{hid}})} \sum_{\alpha'}^{\min(d_{\text{hid}}, d_{\text{in}})} U_{i\alpha}^{(2)} s_\alpha^{(2)} V_{\alpha h}^{(2)\intercal} U_{h\alpha'}^{(1)} s_{\alpha'}^{(1)} V_{\alpha' j}^{(1)\intercal} x_j \\ &= (\boldsymbol{U}^{(2)} \boldsymbol{S}^{(2)} \boldsymbol{V}^{(2)\intercal} \boldsymbol{U}^{(1)} \boldsymbol{S}^{(1)} \boldsymbol{V}^{(1)\intercal} \boldsymbol{x})_i. \end{aligned}$$

If the minimum of weight dimensions $\min(d_{\text{out}}, d_{\text{hid}}, d_{\text{in}})$ exceeds the number of task modes $\sum_m \text{rank}(\boldsymbol{W}^{\star m})$, it is possible to choose/learn $\boldsymbol{V}^{(2)}$ and $\boldsymbol{U}^{(1)}$ such that the second layer singular values $s_\alpha^{(2)}$ effectively take the role of the gates c^p, whereas the first layer encodes the student task representations. If we put aside the question of learnability and only ask about expressivity, this argument shows that a gating structure can emerge as subset of a two-layer network. For the fully-connected model we have in the main text, $d_{\text{hid}} = 2\, d_{\text{out}}$, giving the network the capacity to learn and remember solutions for both teachers.

A.5 Adaptation speed

In this section, we derive the change in model output that is induced by the change in parameters depending on their configuration, thereby describing the model's adaptation speed.

Neural tangent kernel Here, we briefly review the Neural Tangent Kernel (NTK, [Jacot et al., 2020]) which we then use to directly describe the adaptation speed in the output $\boldsymbol{y}(t)$. For a vector-valued model $\boldsymbol{y} \in \mathbb{R}^{d_{\text{out}}}$ parameterized by a flattened parameter vector $\theta^k = (\text{flatten}(W_{ij}^p, c^p))^k$, the output evolves as

$$\frac{\mathrm{d}}{\mathrm{d}t} y_i = \sum_k \frac{\mathrm{d}y_i}{\mathrm{d}\theta^k} \frac{\mathrm{d}\theta^k}{\mathrm{d}t} = -\sum_k \frac{\mathrm{d}y_i}{\mathrm{d}\theta^k} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta^k} = -\underbrace{\sum_{k,j} \frac{\mathrm{d}y_i}{\mathrm{d}\theta^k} \frac{\mathrm{d}y_j}{\mathrm{d}\theta^k}}_{\mathsf{NTK}_{ij}} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}y_j},$$

where we used the chain rule and that the parameters update according to gradient descent $\frac{\mathrm{d}\theta^k}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta^k}$ and have set the learning rate to 1 for simplicity.

This object can be understood as a matrix operating on the output space $\mathsf{NTK} = (\mathrm{d}\boldsymbol{y}/\mathrm{d}\theta^\intercal)\,(\mathrm{d}\boldsymbol{y}^\intercal/\mathrm{d}\theta) \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$, where the inner product represents the sum across parameters $\sum_k$ in the expression above. For the reduced model Eq. (2) with $\theta = \text{flatten}(c^p, w_i^p)_{p=1\ldots P}$, we readily get

$$\frac{\mathrm{d}y_i}{\mathrm{d}c^p} = w_i^p, \quad \frac{\mathrm{d}y_i}{\mathrm{d}w_j^p} = \delta_{ij}\, c^p,$$

where δ_{ij} is the Kronecker delta.

We then arrive at

$$\mathsf{NTK} = \sum_p c^p c^p + \boldsymbol{w}^p \boldsymbol{w}^{p\intercal}, \tag{20}$$

where we adopt standard matrix notation to imply $c^p c^p \equiv c^p c^p\, \boldsymbol{I}_{d_{\text{out}}}$ as being proportional to the identity matrix $\boldsymbol{I}_{d_{\text{out}}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$.

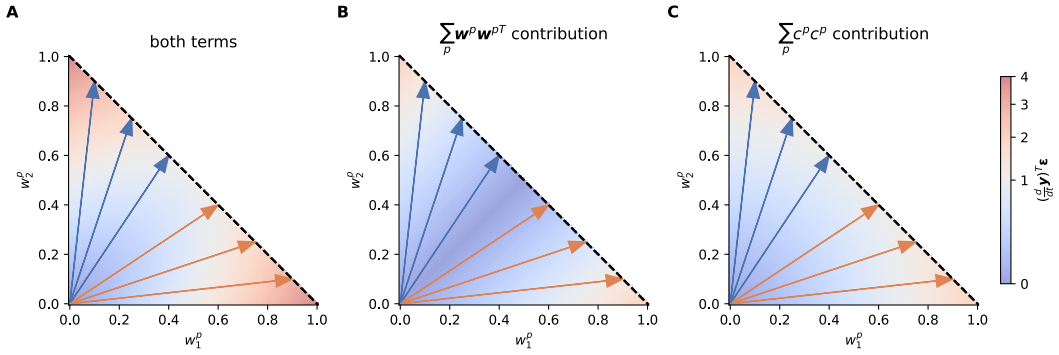

Figure A.8: Specialized students and gates accelerate adaptation. Heatmaps of the dot product $(\frac{d}{dt}\boldsymbol{y})^\intercal \boldsymbol{\varepsilon}$ contributions for different terms of the Neural Tangent Kernel (NTK), depending on specialization of weight vectors $\boldsymbol{w}^1$ (*blue*), $\boldsymbol{w}^2$ (*orange*), of which three pairs corresponding to different degrees of specialization are shown here (pairs are formed by vectors that are symmetric along the diagonal). c^1, c^2 are scaled so that the sum lies on the *dashed black line* (given by L^1 regularization). **A.** shows the total contribution of both terms of Eq. (20) combined, **B.** isolates the contribution from the $\boldsymbol{w}^p \boldsymbol{w}^{p\intercal}$ term, and **C.** displays the contribution from the $c^p c^p$ term. Dashed lines indicate possible solutions.

Accelerated adaptation through specialized weights and selective gates To study the accelerated adaptation of the loss $\mathcal{L}_{\text{task}} = {}^1\!/_2\,(\boldsymbol{y}^{\star m} - \boldsymbol{y})^2$, we use the Neural Tangent Kernel of the architecture that directly describes the dynamics of the model output. To this end, we study how the model output $\boldsymbol{y}$ changes in response to a block switch $\boldsymbol{y}^{\star m} = (1,0)^\intercal \to (0,1)^\intercal$, entailing an error $\boldsymbol{\varepsilon} = (-1,1)^\intercal$

that drives a change in model output. To see how this change reduces the loss, we calculate its alignment with the error term as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}_{\text{task}} = \boldsymbol{\varepsilon}^{\intercal}\left(\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{y}\right) &= \boldsymbol{\varepsilon}^{\intercal}\,\mathsf{NTK}\,\boldsymbol{\varepsilon} \\ &= \boldsymbol{\varepsilon}^{\intercal}\sum_{p}\left[c^{p}c^{p} + \boldsymbol{w}^{p}\boldsymbol{w}^{p\intercal}\right]\boldsymbol{\varepsilon} \\ &= \sum_{p}\|\boldsymbol{\varepsilon}\|^{2}\left(c^{p}\right)^{2} + \left(\boldsymbol{w}^{p\intercal}\boldsymbol{\varepsilon}\right)^{2}.\end{aligned}$$

The NTK reveals that the change in output $\boldsymbol{y}$ is accelerated through two contributions which we illustrate in Fig. A.8: First, we observed that student-teacher alignment which enters $|\boldsymbol{w}^{p\intercal}\boldsymbol{\varepsilon}|$ increases towards the flexible regime. We note that this acceleration however does not require unique student-teacher alignment (such that no two students match the same teacher); it is the joint effect of asymmetric gates which further facilitates unique specialization. Second, selective gates accelerate adaptation because a sparse vector with the same norm tends to have a larger sum-of-squares $\sum_p (c^p)^2$ that enters the NTK. These factors coincide in the flexible regime.

A.6 Larger blocks enable faster specialization

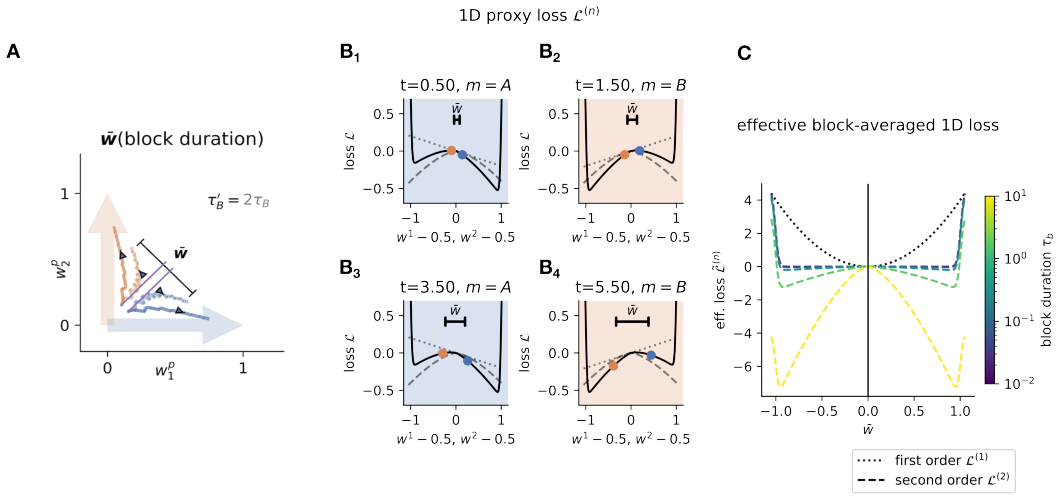

Figure A.9: Larger blocks enable faster specialization. **A.** Two trajectories of differing block length (*faint*: τ_B, *opaque*: $\tau'_B = 2\tau_B$) of two students (*blue*, *orange*) in teacher subspace as in Fig. 4. **B.** Student dynamics in an approximate loss landscape in early learning. Subpanels 1-4 are time points in a simulation. *Background*: active context m. The linear first-order loss does not lead to separation, as a block switch will exactly reverse any changes to specialization. In contrast, the curvature from the second order term enables students to accumulate an initial advantage in specialization. **C.** Like **B.**, but loss is in terms of the specialization variable $\bar{w}$. The effective loss over blocks depends on block size τ_B: the longer the τ_B is, the longer the students will have to fall down the landscape in **B.**. The first-order term, corresponding to infinitely short blocks, does not prefer specialization.

Here, we calculate how the block length affects the specialization $\bar{\boldsymbol{w}} \coloneqq \boldsymbol{w}^1 - \boldsymbol{w}^2$ in students, given equal total learning time t. To do so, we consider periods early in learning where the students have not specialized yet, $\bar{\boldsymbol{w}} \simeq 0$, and the gates consequentially are indifferent, $\bar{\boldsymbol{c}} = 0$. We then consider a

period of the task, i.e. back-and-forth block switches $a \to b$, $b \to a$ that last a total of $T = 2 \cdot \tau_B$. We analyze the limit of small block sizes and ask how $\bar{\boldsymbol{w}}$ changes over T: for short blocks, the model is time-reversal symmetric, i.e. any change $(\frac{\mathrm{d}}{\mathrm{d}t}\bar{w})\,\tau_B$ during $a \to b$ is exactly undone during $b \to a$. We therefore calculate the second-order effects to $\bar{\boldsymbol{w}}$ to analyze the dependence on τ_B, where we only make an assumption on the approximate directions of $\boldsymbol{w}$ and $\boldsymbol{\varepsilon}$ that apply early in learning, but leave their scale general.

Second-order derivatives for weight and gates To prepare, we first calculate the second order derivatives of the updates which we will need in what follows. By application of the product rule, we obtain for the weights $\boldsymbol{w}^p$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{w}^p &= c^p\boldsymbol{\varepsilon},\\ \frac{\mathrm{d}^2}{\mathrm{d}t^2}\boldsymbol{w}^p &= \left(\frac{\mathrm{d}}{\mathrm{d}t}c^p\right)\boldsymbol{\varepsilon} + c^p\left(\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\varepsilon}\right)\\ &= \boldsymbol{w}^{p\mathsf{T}}\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon} - c^p\,\mathsf{NTK}\,\boldsymbol{\varepsilon}\\ &= \boldsymbol{w}^{p\mathsf{T}}\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon} - c^p(\textstyle\sum_{p'} c^{p'}c^{p'} + \boldsymbol{w}^{p'}\boldsymbol{w}^{p'\mathsf{T}})\boldsymbol{\varepsilon}\end{aligned}$$

where we use $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\varepsilon} = \frac{\mathrm{d}}{\mathrm{d}t}\,(\boldsymbol{y}^{\star m} - \boldsymbol{y}) = -\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{y} = \frac{\mathrm{d}}{\mathrm{d}t}\mathsf{NTK}\,\boldsymbol{\varepsilon}$ within a block.

For the gates c^p , we get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}c^p &= \boldsymbol{w}^{p\mathsf{T}}\boldsymbol{\varepsilon},\\ \frac{\mathrm{d}^2}{\mathrm{d}t^2}c^p &= c^p\boldsymbol{\varepsilon}^{\mathsf{T}}\boldsymbol{\varepsilon} - \boldsymbol{w}^{p\mathsf{T}}\,\mathsf{NTK}\,\boldsymbol{\varepsilon}.\end{aligned}$$

We then take the difference of the weight derivatives $\bar{\boldsymbol{w}} = \boldsymbol{w}^1 - \boldsymbol{w}^2, \bar{c} = c^1 - c^2$ to get by linearity

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\bar{\boldsymbol{w}} &= \bar{c}\boldsymbol{\varepsilon} \;\to\; 0\\ \frac{\mathrm{d}^2}{\mathrm{d}t^2}\bar{\boldsymbol{w}} &= \bar{\boldsymbol{w}}^{\mathsf{T}}\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon} - \bar{c}\,(\textstyle\sum_{p'} c^{p'}c^{p'} + \boldsymbol{w}^{p'}\boldsymbol{w}^{p'\mathsf{T}})\boldsymbol{\varepsilon} \;\to\; \bar{\boldsymbol{w}}^{\mathsf{T}}\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon}\end{aligned}$$

where we assume that the gates have approximately equal values $\bar{c} \approx 0$ when no specialization has taken place yet.

Effect on specialization The sum of second derivatives after having seen a switch $a \to b$ and $b \to a$ is subsequently:

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\bar{\boldsymbol{w}}\Big|_{a\to b} + \frac{\mathrm{d}^2}{\mathrm{d}t^2}\bar{\boldsymbol{w}}\Big|_{b\to a} &= \bar{\boldsymbol{w}}^{\mathsf{T}}\left(\boldsymbol{\varepsilon}^b\,\boldsymbol{\varepsilon}^b + \boldsymbol{\varepsilon}^a\,\boldsymbol{\varepsilon}^a\right)\\ &= \|\bar{\boldsymbol{w}}\|\,\|\boldsymbol{\varepsilon}\|\left(\boldsymbol{\varepsilon}^b - \boldsymbol{\varepsilon}^a\right)\\ &= 2\,\|\bar{\boldsymbol{w}}\|\,\|\boldsymbol{\varepsilon}\|\;\boldsymbol{\varepsilon},\end{aligned}$$

where we use that $\bar{\boldsymbol{w}}$ is the component parallel with the error signal, implying $\bar{\boldsymbol{w}}^{\mathsf{T}}\boldsymbol{\varepsilon}^p = \|\bar{\boldsymbol{w}}\|\,\|\boldsymbol{\varepsilon}^p\|$ for the errors $\boldsymbol{\varepsilon}^b, \boldsymbol{\varepsilon}^a = \pm 1/2\, C\,(1, -1)^{\mathsf{T}}$ for some constant C depending on weight magnitude, and small $\bar{\boldsymbol{w}}(0)$. We herein assumed that the errors between blocks only differ by a sign, $\boldsymbol{\varepsilon} \coloneqq \boldsymbol{\varepsilon}^b = -\boldsymbol{\varepsilon}^a$. In particular, we neglect the change in c^p for simplicity. Note that this is a good approximation only if the block size τ_B is much shorter than the timescales of τ_c. While this limits quantitative predictions of this approximation for the setting we consider, we expect that it identifies the qualitative mechanism.

Introducing the period T, which is double the block length $\tau_B = T/2$ (spanning two blocks of length τ_B), we get

$$\bar{\boldsymbol{w}}(T = 2 \cdot \tau_B) = \bar{\boldsymbol{w}}(0) + 2 \cdot \frac{1}{2} \left(2 \left\|\bar{\boldsymbol{w}}\right\| \left\|\boldsymbol{\varepsilon}\right\| \boldsymbol{\varepsilon}\right) \tau_B^2 \tag{21}$$

where the factor $\frac{1}{2}$ in the first line is due to being at second order in the Taylor expansion of the update. Setting $\bar{\boldsymbol{w}}(0) = 0$, this means that the cumulative change of two periods together lasting $t = 2T$ (i.e. spanning two pairs of blocks totaling four blocks) is

$$\bar{\boldsymbol{w}}(t \;=\; 2 \text{ periods of } T \;=\; 2 \times (2 \cdot \tau_B))) = 2 \times 2 \cdot \frac{1}{2} \left(2 \left\|\bar{\boldsymbol{w}}\right\| \left\|\boldsymbol{\varepsilon}\right\| \boldsymbol{\varepsilon}\right) \tau_B^2 = 2 \left(2 \left\|\bar{\boldsymbol{w}}\right\| \left\|\boldsymbol{\varepsilon}\right\| \boldsymbol{\varepsilon}\right) \tau_B^2 \tag{22}$$

In contrast, doubling the block size $\tau_B' = 2\tau_B$ thus ($T' = 2T$), but running for the same amount of time t (i.e., two blocks of double the block length τ_B) gives

$$\bar{\boldsymbol{w}}(t \;=\; 1 \text{ period of } T' \;=\; 1 \times (2 \cdot 2\tau_B)) = 1 \times 2 \cdot \frac{1}{2} \left(2 \left\|\bar{\boldsymbol{w}}\right\| \left\|\boldsymbol{\varepsilon}\right\| \boldsymbol{\varepsilon}\right) \left(2\tau_B\right)^2 = 4 \left(2 \left\|\bar{\boldsymbol{w}}\right\| \left\|\boldsymbol{\varepsilon}\right\| \boldsymbol{\varepsilon}\right) \tau_B^2,$$

which is twice as large as the short-block version. This explains why larger blocks can lead to faster specialization.

Mechanical analogy The importance of the quadratic term for specialization can be understood through a mechanical analogy for the weights as particles: gradient flow corresponds to the dynamics of particles undergoing an overdamped Newtonian motion in a potential. To this end, we consider the proxy loss potential $\mathcal{L}^{(n)}$ that induces the respective first- and second-order time dynamics described by Eq. (21) when considering gradient flow $\tau_w \frac{\mathrm{d}}{\mathrm{d}t} w = -\nabla_w \mathcal{L}^{(n)}(w)$. The resulting loss potential $\mathcal{L}^{(n)}$ is polynomial and grows $\propto -w$ (first-order) and $\propto -|w|^{3/2}$ (second-order), as can be verified by plugging in a solutions $w(t) \propto t$, $w(t) \propto t^2$.

To first-order, block changes will result in exactly opposite gradients, which will revert any changes from the previous block due to the lack of momentum effects in gradient flow (Fig. A.9B, *dotted line*). In contrast, the quadratic term in the time dynamics can be understood as resulting from an effective non-linear loss potential, breaking this time-reversal symmetry between blocks (Fig. A.9B, *dashed line*). Note that the sum $\mathcal{L}^{(1)} + \mathcal{L}^{(2)}$ may still be monotonic.

When aggregating this effect over many block changes, it gives rise to an *effective* loss potential $\tilde{\mathcal{L}}$ for the specialization variable $\bar{w}$ (Fig. A.9C). As the first-order terms cancel out over blocks, the effective loss potential does not have a preferred specialized configuration. When including second-order terms however, the preferred state becomes specialized. Moreover, the speed of this specialization depends on the timescale of the blocks τ_B: the larger τ_B, the further the particles increase their advantage down the non-linear loss potential in Fig. A.9B, which manifests as a steeper and thereby faster double-well loss potential in Fig. A.9C.

A.7 Exact solutions under the condition of symmetry

A.7.1 Solving the differential equations under the condition of symmetric specialization

The multiplicatively-coupled dynamics allow for emergent specialization of the students for one of the teachers. Due to the regularization on c as well as the orthogonality of the teachers, these dynamics are competitive, meaning that the students will, after a number of task switches, each specialize for a different teacher. We saw before that the most rapid adaptation occurs when the system is in this state where each teacher is matched by exactly one student. We formalize this state more generally under a condition of *symmetric specialization*, in which $\bar{w} = \bar{w}_1 = w_1^1 - w_1^2 = w_2^2 - w_2^1 = \bar{w}_2$. In this section, we present exact solutions to the learning dynamics that occur in this system under this condition. We assume without loss of generality that path p in the NTA student is aligning with teacher m. We will see that these solutions closely match the adaptation dynamics of the full NTA model.

The learning updates of the two specialization components are $\frac{\mathrm{d}\bar{w}_1}{\mathrm{d}dt} = \bar{c}\,\varepsilon_1$ and $\frac{\mathrm{d}\bar{w}_2}{\mathrm{d}dt} = -\bar{c}\,\varepsilon_2$. Therefore, the symmetric specialization condition is inherently preserved when $\varepsilon_1 = -\varepsilon_2$. We calculate the accuracy of both relationships in Fig. A.10.

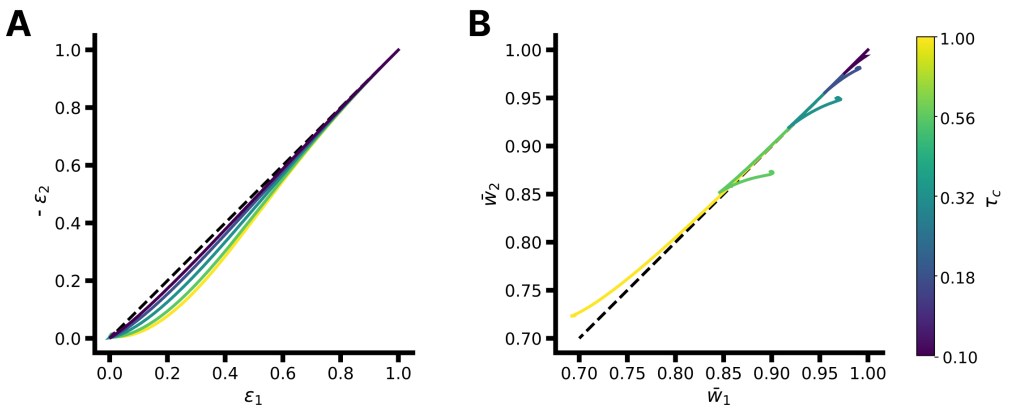

Figure A.10: Accuracy of assumptions used in calculating the exact solutions, only one of which is required. A. Accuracy of the assumption $\bar{\varepsilon} = \varepsilon_1 = -\varepsilon_2$. **B.** Accuracy of the assumption $\bar{w} = w_1^1 - w_1^2 = w_2^2 - w_2^1$.

A.7.2 Learning occurs dominantly within but also outside of the specialization subspace over the course of a single block

The differential equation Eq. (9) then results from dividing the two update equations for $\frac{\mathrm{d}}{\mathrm{d}c}$ and $\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}}$. We can then solve this differential equation to obtain the relationship

$$\tau_c \, \bar{c}^2 = 2 \, \tau_w \, \bar{w}^2 + C \tag{23}$$

where C is an integration constant, which we determine by plugging in the initial conditions $\bar{c} = \bar{w} = 1$ to represent the theoretical ideal for the flexible regime.

A.7.3 Learning in the orthogonal component

As is highlighted by the fact that the analytical solution is not traversed in full in Fig. 4E, learning also occurs outside of the specialization subspace. This learning can be characterized by co-specialization which is characteristic of the forgetful regime

$$\bar{\bar{w}} := \frac{(w_1^1 - w_2^1) + (w_1^2 - w_2^2)}{2}. \tag{24}$$

Fig. A.11 shows learning along both components $\bar{w}$ and $\bar{\bar{w}}$, the two error components ε_1 and ε_2, as well as the separation of the gates $\bar{c}$ over the course of a single block beginning from the same initial conditions used in the differential equation.

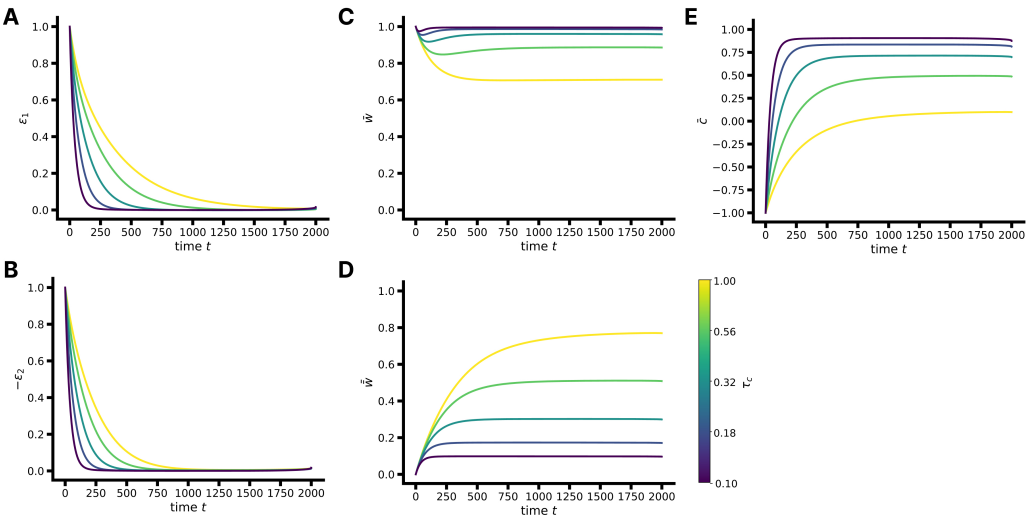

Figure A.11: Learning occurs within and outside of the specialization subspace. A. First component of the error following a task switch from task A to task B for different values of τ_c. **B.** Second component of the error across the same timeframe. **C.** Adaptation of weight matrices in the specialization space. **D.** Orthogonal component of learning that measures adaptation of both teachers for the current task. **E.** Gate change in the specialization subspace.

A.8 Gated model generalizes to perform task and subtask composition

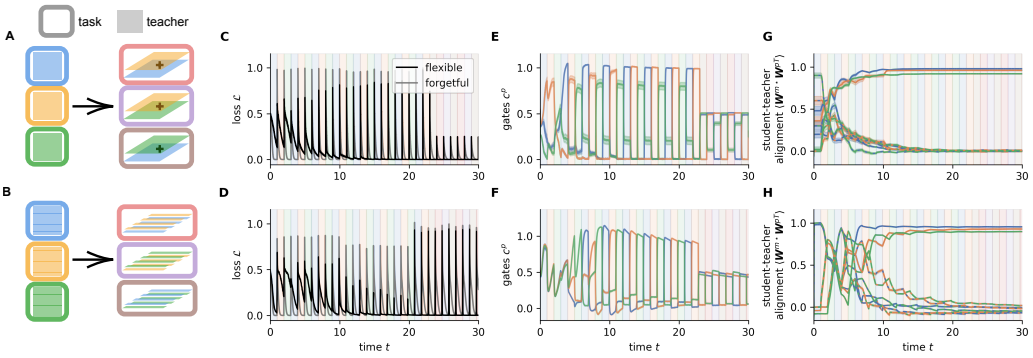

Figure A.12: Gated model generalizes to compositional tasks. A. Task composition consists of new tasks that sum sets of teachers previously encountered. **B.** Subtask composition consists of new tasks that concatenate alternating rows of sets of teachers previously encountered. Loss (**C.**,**D.**), gating activity (**E**,**F.**), and student-teacher alignment (**G.**,**H.**) of models on generalization to task composition (*top*) and subtask composition (*bottom*).

As stated in the main text, we consider two settings to evaluate whether the gate layer can recombine previous knowledge for compositional generalization. We first train the NTA model with three paths on three teachers A, B, and C individually, and then change the network on **task composition** (Fig. A.12A) or **subtask composition** (Fig. A.12B). Task composition proceeds the same as our standard setup.

In subtask composition, to allow our model the possibility to compose not only tasks (i.e., gate the entire student matrix $\boldsymbol{W}^p$), but also subtasks (individual rows of $\boldsymbol{W}^p$), we increase the expressiveness of our gating layer by using an independent gate for each neuron (or row of $\boldsymbol{W}^p$) in the student hidden layer and allow gradient descent to update these gates individually. We call this the per-neuron gating version of our model.

In principle, the per-neuron gating NTA has Pd_{out} independent paths modulated by gates. Thus, in order to study whether specialization and gating occurs for each teacher, we sort the Pd_{out} paths into P paths of size d_{out}. We do this by computing the cosine similarity between each row in the first layer $\boldsymbol{W}$ and the teachers $\boldsymbol{W}^\star$. We then sort the rows of the first layer to align with the rows of the rows of the teachers that they best match, such that we identify their respective students to visualize student-teacher alignment (Fig. A.12H). Additionally, we permute the gating layer $\boldsymbol{c}$ to match this sorting. We take the mean of the sorted gates for each student to visualize the task-specific gating (Fig. A.12F).

We find that both the per-student and per-neuron gating NTA models can solve task and subtask generalization tasks, respectively, and maintain their learned specialization after transitioning to compositional settings (Fig. A.12C-H). We additionally observe that the gating variables learn to appropriately match the latent structure of the generalization tasks by turning off the non-contributing gate and evenly weighting the two "on" gates. We again observe the rapid adaptation of the flexible NTA to compositional tasks compared to the forgetful regime.

A.9 Non-orthogonal teachers

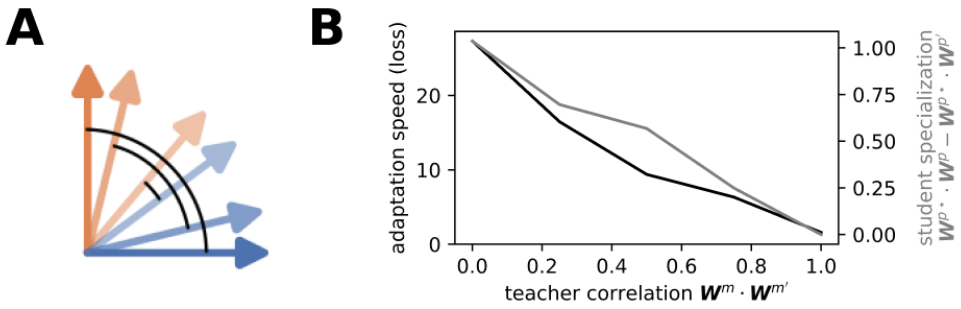

Figure A.13: Robustness to relaxing orthogonality between teachers. A. Illustration of changing teacher cosine similarity. **B.** Adaptation speed as measured by the loss after a block switch (*black*) and student specialization (*gray*), both as a function of the teacher similarity. 0 represents the orthogonal case studied in the main text.

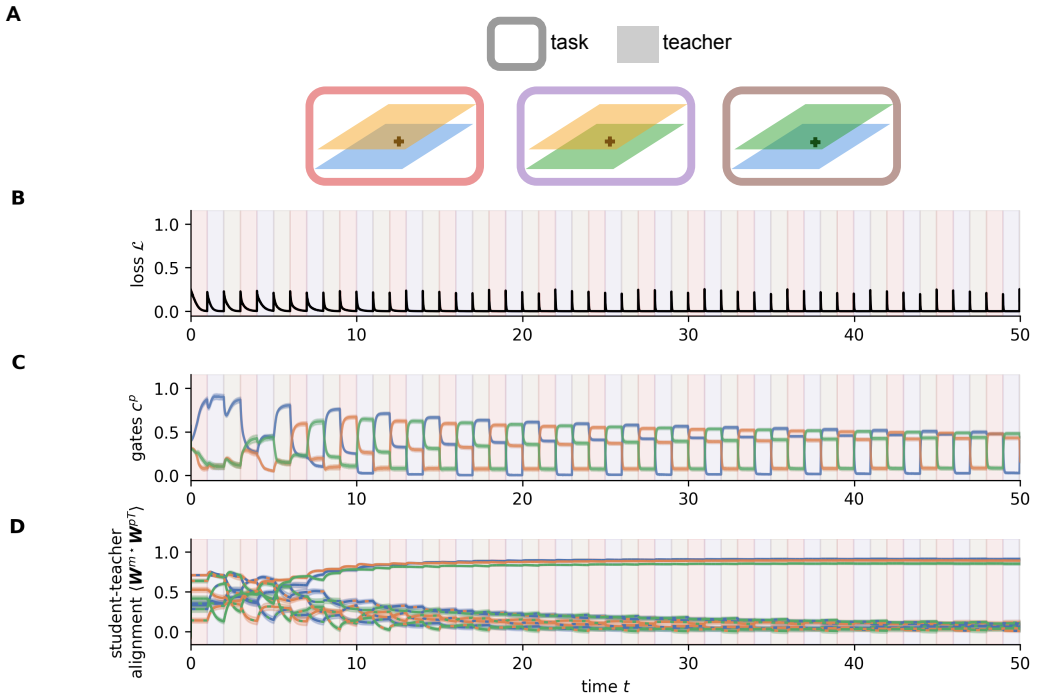

Figure A.14: Flexible NTA successfully specializes to underlying teachers even when trained on non-orthogonal tasks. A. Tasks are created by adding different pairs of teachers, such that each task is non-orthogonal to every other task. **B.** Loss during learning. **C.** Gating variables learn to appropriately match the latent structure of the tasks. **D.** Students learn to specialize to teacher components, despite the non-orthogonality of the tasks.

In the main text, we have worked with the assumption that different tasks are approximately orthogonal to permit our theoretical analysis. This assumption holds for randomly-generated teachers when the input dimension is high. In simulations, we implemented this condition by constructing the corresponding rows of teachers to be orthogonal, $\boldsymbol{w}_i^{\star 1} \cdot \boldsymbol{w}_i^{\star 2} = 0$. Still, it is unclear what happens when the teachers are not even approximately orthogonal. We investigate this question empirically in Fig. A.13 and find that specialization decays gracefully as the orthogonality assumption is relaxed.

We also design a set of three non-orthogonal tasks using three orthogonal teachers, where each task is created by adding different pairs of the teachers (Fig. A.14A). Thus, every task has some similarity (and is non-orthogonal) with every other task. We find that the flexible NTA can successfully solve these tasks and identify their underlying latent teacher structure, learning to specialize and gate all teacher components which comprise the overall set of tasks (Fig. A.14B-D).

A.9.1 Experiments on fashionMNIST dataset

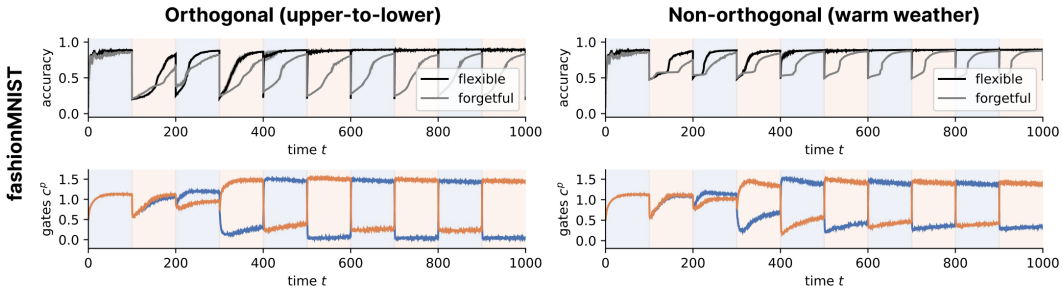

Figure A.15: NTA quickly adapts across fashionMNIST for (*left*) an orthogonal sorting based on upper-to-lower items of clothing and (*right*) a correlated sorting for warm-to-cold weather clothing. The panels show (*top*) accuracy on the test set and (*bottom*) activity of the gates. We show mean and standard error with 10 seeds.

We explicitly compare the network's performance on two different versions of fashionMNIST based on tasks that might appear in a real-world setting. The original fashionMNIST dataset has items sorted roughly by order of commonality, with the label 0 being assigned to T-shirts, and the label 9 being assigned to ankle boots. We generate two different permutations of these labels representing other real-world sorting of the items that have different amounts of shared structure with the original. The close-to-orthogonal ordering sorts the clothing from upper to lower body, and orders the labels 0, 2, 4, 6, 8, 1, 3, 5, 7, 9. The ordering with more shared structure represents warm to cold weather clothing, and orders the labels to 0, 1, 5, 3, 7, 6, 2, 4, 8, 9. The results show that the stereotypical NTA-like task switching behavior and specialization emerges for both settings at a similar speed despite baseline performance being higher on the task with shared structure (Fig. A.15).

A.10 Few-shot adaptation in the low sample rate regime

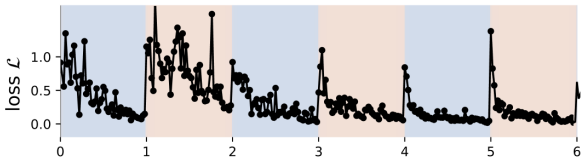

Figure A.16: Few-shot adaptation after block switches. Like Fig. 2A in the main text, but with coarsely discretized time to examine the adaptation after a single sample. As this drastically reduces signal-to-noise ratio, we average over 100 samples. Markers indicate a single step of gradient descent on one sample.

In the main text, we have considered the case of large batch size (or equivalently, small time discretization) that allows taking a sample average when going towards the theoretical, equivalent model. This averaging reduces noise in the gradient signal stemming from random samples, so that it is unclear whether learning is still possible when sample rate is low. As theoretical analysis is challenging for this case, we investigate this empirically in Fig. A.16 and find that the qualitative phenomenon is preserved even if only a single sample $B = 1$ is used for every gradient update.

B Technical details

B.1 Notation

Table 1: Overview of notation used throughout the paper.

Symbol	Description
$\boldsymbol{x}^b \in \mathbb{R}^{d_{\text{in}}},\ b = 1...B$	input sample of a batch
$\boldsymbol{y}^b \in \mathbb{R}^{d_{\text{out}}},\ b = 1...B$	model output
$\boldsymbol{y}^{\star m} \in \mathbb{R}^{d_{\text{out}}},\ m = 1...M$ or $a, b, ...$	target label in task m
$\boldsymbol{\varepsilon} = \boldsymbol{y}^\star - \boldsymbol{y} \in \mathbb{R}^{d_{\text{out}}}$	prediction error
$c^p \in \mathbb{R},\ p = 1...P$	gates for each pathway p
$\boldsymbol{W}^p \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$	student weights for each pathway p
$\boldsymbol{W}^{\star m} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}},\ m = 1...M$ or $a, b, ...$	teacher weights for each task m
$\boldsymbol{w}^p \equiv \boldsymbol{w}^p_\alpha \in \mathbb{R}^2$	2D vector for reduced model (for each teacher singular value α)
$W_{ij} = \sum_\alpha^{\min(d_{\text{out}}, d_{\text{in}})} U_{i\alpha} s_\alpha V^\intercal_{\alpha j}$	singular value decomposition of weights
$\tau_w = \eta_w^{-1}, \tau_c = \eta_c^{-1}$	parameter time scale (inverse learning rate)
τ_B	block length
$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}}$	loss
$\bar{w}_1 = w^{p=1}_{m=1} - w^{p=2}_{m=1}$	specialization for teacher 1
$\bar{w}_2 = w^{p=2}_{m=2} - w^{p=1}_{m=2}$	specialization for teacher 2
$\bar{w} = \frac{1}{2}(\bar{w}_1 + \bar{w}_2)$	overall specialization
$\bar{c} = c^1 - c^2$	separation of gates
$\bar{\bar{w}} = ((w^{p=1}_{m=1} - w^{p=1}_{m=2}) + (w^{p=2}_{m=1} - w^{p=2}_{m=2}))/2$	unspecialized learning

B.2 Hyperparameters

We perform the gradient calculations and the simulation of gradient flow using the JAX framework and make our implementation publicly available at https://github.com/aproca/neural_task_abstraction.

Our hyperparameters for all experimental settings are listed in the table below. We use $\mathcal{L}_{\text{norm-L1}}$ for most experiments, except for generalization to subtask composition and the flexible fully-connected network experiments where we use $\mathcal{L}_{\text{norm-L2}}$ (see Appendix B.3 for a discussion on this choice). For the cases where we induce the forgetful regime as an experimental control, we use the same hyperparameters, set regularization to 0 ($\lambda_{\text{nonneg}}, \lambda_{\text{norm-L1}}, \lambda_{\text{norm-L2}} = 0$), and may or may not adjust the learning rate of the gating layer. Differences in hyperarameters from the main model and control are denoted in the tables below as 'main / control.'

Table 2: Hyperparameters.

Hyperparameter	Task specialization (Fig. 2)	Task composition (Fig. 3,A.12)	Subtask composition (Fig. 3,A.12)	Reduced model (Fig. 4)	Fully-connected network (Fig. 6,A.4,A.5,A.6)	MNIST (Fig. 7)
P	2	3	3	2	2	2
M	2	3	3	2	2	2
d_{in}	20	20	20	1	20	64
d_{hid}					20	
d_{out}	10	6	6	2	10	10
λ_{nonneg}	0.091 / 0	0.5 / 0	0.023 / 0	0.091 / 0	0.2 / 0	0.5 / 0
$\lambda_{\text{norm-L1}}$	0.456 / 0	1.25 / 0	0	0.455 / 0	0	0.25 / 0
$\lambda_{\text{norm-L2}}$	0	0	0.011 / 0	0	0.1 / 0	0
τ_w	1.3	0.2	0.2	5	0.06	10
τ_c	0.03 / 1.3	0.03	0.005	0.7	0.01	0.005 / 10
batch size	200	200	200	200	200	100
seeds	10	10	10	1	10	10
number of blocks n	20	30	30	17	30	10
τ_B	1	1	1	1	1	1
dt	0.001	0.001	0.01	0.001	0.01	0.001

Table 3: Hyperparameters II.

Hyperparameter	NTA hyperparameter search (Fig. 5)	Fully-connected hyperparameter search (Fig. A.7)	Task switching (Fig. 8)	Non-orthogonal tasks (Fig. A.14)	Non-orthogonal teachers (Fig. A.13)
P	2	2	2	3	2
M	2	2	2	3	2
d_{in}	20	20	20	20	20
d_{hid}		20			
d_{out}	10	10	10	6	10
λ_{nonneg}	0.5	0.23	0.18 / 0	0.33	0
$\lambda_{\text{norm-L1}}$	1.25	0	0.36 / 0	0.83	0
$\lambda_{\text{norm-L2}}$	0	0.11	0	0	0.5
τ_w	0.1	0.04	0.07	0.05	0.016
τ_c	0.005	0.01	0.01	0.03	0.016
batch size	200	200	200	200	200
seeds	10	10	10	10	1
number of blocks n	7	20	30	50	10
τ_B	1	1	1	1	1
dt	0.001	0.01	0.01	0.001	0.01

Table 4: Hyperparameters III.

Hyperparameter	Full vs. reduced model (Fig. A.1)	Slow high-d students (Fig. A.2)	Redundant paths (Fig. A.3)	Few-shot adaptation (Fig. A.16)	fashionMNIST (Fig. A.15)
P	2	2	4	2	2
M	2	2	2	2	2
d_{in}	20 / 1	30	20	20	64
d_{out}	10 / 2	30	10	10	10
λ_{nonneg}	0.091	0.091	0.194 / 0.545	0.091	0.5 / 0
$\lambda_{\text{norm-L1}}$	0.455	0.455	0.968 / 2.727	0	0
$\lambda_{\text{norm-L2}}$	0	0	0	0.455	0.25 / 0
τ_w	1.3	0.5	1.3	1	10
τ_c	0.03 / 0.06	0.1	0.03	0.01	0.005 / 10
batch size	200	200	200	1	100
seeds	1	10	1	10	10
number of blocks n	20	20	20	6	10
τ_B	1	1	1	1	1
dt	0.001	0.001	0.001	0.02	0.001

B.3 Regularization

We use a combined regularizer that is motivated by biological constraints on our model parameters. The regularizers alleviate the underspecification of the solution space of our linear model and facilitate symmetry breaking to allow the model to specialize different components, while not forcing specialization (Fig. B.1). Here, we detail the effect of these regularizing terms. To this end, recall the definition of the reduced model, Eq. (14)

$$\boldsymbol{y} = c^1 \boldsymbol{w}^1 + c^2 \boldsymbol{w}^2.$$

Nonnegative neural activity The gating variables steer the model output multiplicatively. Biologically, such an interaction is mediated by a firing rate, an inherently positive variable. Computationally, this has implications on the solution space: with random initialization, almost all configurations of $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ form a basis of $\mathbb{R}^2$. By definition, this means that there will always be two coefficients c^1, $c^2 \in \mathbb{R}$ that will yield the correct solution. Nonnegativity constrains this set to lie in the positive quadrant of a 2D space. In particular,

$$\mathcal{L}_{\text{nonneg}} = \sum_{p=1}^{P} \max(0, -c^p)$$

Alleviating invariance via competition Even in the desired specialized configuration, the model is invariant under

$$c^p, \boldsymbol{W}^p \to ac^p, \boldsymbol{W}^p/a$$

for any scalar a. We hence bound the norm of the vector $\boldsymbol{c} = \left(c^1, c^2\right)^\intercal$.

We consider two regularizers based on the L^1 and L^2 norm

$$\mathcal{L}_{\text{norm-L1}} = {}^1\!/_2(\|\boldsymbol{c}\|_1 - 1)^2$$
$$\mathcal{L}_{\text{norm-L2}} = {}^1\!/_2(\|\boldsymbol{c}\|_2 - 1)^2$$

$\mathcal{L}_{\text{norm-L1}}$ encourages sparsity in the gates which is beneficial when there are few active gating variables as in the NTA model (with a single teacher active in each task). However, if we consider cases with potentially many active gating variables, as in the per-neuron gating NTA (see Appendix A.8) or a deep fully-connected network (or even many teachers active at once), favoring sparsity restricts expressivity of the model. In these cases, we instead use $\mathcal{L}_{\text{norm-L2}}$. In practice, both regularizers facilitate specialization robustly across many settings.

Applying nonnegativity and norm regularization together has the effect of inducing competition between gating variables. While there is a solution where gates are equal in magnitude (as shown in Fig. B.1), deviating from this solution while minimizing regularization loss will cause one gate to increase in magnitude and the other to decrease. Thus, this competitive effect facilitates symmetry-breaking in the gates.

Finally, we note that although these regularizers facilitate symmetry breaking through competition, they simultaneously allow for compositionality such that multiple gates can be active at once. We show this in several experiments, namely our studies of nonorthogonal tasks (Appendix A.9), compositional generalization (Appendix A.8), and fully-connected networks (Appendix A.4).

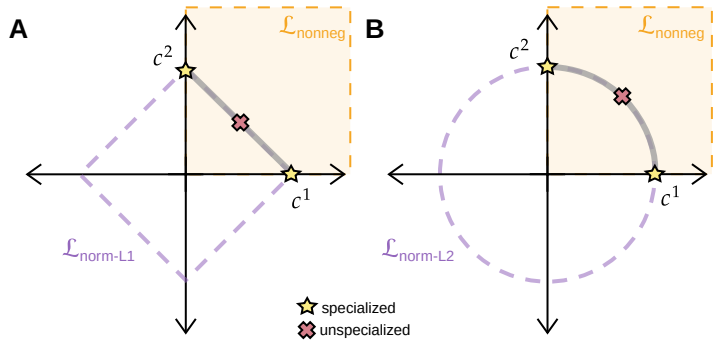

Figure B.1: The effect of regularization on gating variables. Regularization encourages competition between gates while preventing degeneracy of solutions. Importantly, regularization does not force gating variables to be specialized, as illustrated by the red $\times$. This holds for two regularizers we consider, **A.** $\mathcal{L}_{\text{norm-L1}}$ and **B.** $\mathcal{L}_{\text{norm-L2}}$.

B.4 Description of metrics used across experiments

Student-teacher alignment We compute a metric of alignment of each student and each teacher to determine whether students are specializing and, if so, to which teachers they specialize. We do this by computing the similarity between each student $\boldsymbol{W}^p$ and teacher $\boldsymbol{W}^{\star m}$. More specifically, we take the mean of the cosine similarity between student and teacher row vectors. We then sort each student and its gate c^p to the teacher it has the highest cosine similarity with.

Total alignment We compute a metric of total alignment of the network students and teachers to evaluate overall specialization. After computing student-teacher alignment and sorting each student to its respective teacher, we concatenate all students and teachers and compute the overall cosine similarity between the set of students and teachers.

B.4.1 Description of sorting performed in per-neuron NTA and deep fully-connected network

In the cases of more expressive models, such as the per-neuron NTA and fully-connected network, there are Pd_{out} and $d_{\text{out}} \times Pd_{\text{out}}$ respective independent paths modulated by gates. Thus, in order to study whether specialization and gating occurs for each teacher, we sort these into P paths. To do this, we compute the cosine similarity between each row in the first layer $\boldsymbol{W}$ and each row in the teachers $\boldsymbol{W}^\star$. We then sort the rows of the first layer to align with the rows of the teachers that they best match.

Additionally, we permute the second layer to match this sorting. In the per-neuron NTA, this corresponds to scalar gates that are multiplied to each row. In the fully-connected network, this corresponds to the columns of the second layer. Finally, we take the mean of the sorted gates (set of d_{out} columns for the fully-connected network) for each student to visualize teacher-specific gating.

B.5 Hyperparameter search

NTA For the two hyperparameter searches we perform, we run the NTA model on each set of hyperparameters and report the total alignment of concatenated teachers and students at the end of training as an overall measure of specialization. We fix all other hyperparameters.

When varying gate learning rate and block length, we fix the regularization strength to $\lambda_{\text{nonneg}} = 0.5$, $\lambda_{\text{norm-L1}} = 1.25$. When varying regularization strength, we fix gate learning rate $\tau_w/\tau_c = 20$. The regularization strength λ is multipled separately for each type of regularizer such that $\lambda_{\text{nonneg}} = 5\lambda/3$ and $\lambda_{\text{norm-L1}} = 25\lambda/6$.

Fully-connected network We also perform two hyperparameter searches on the fully-connected network. We run the fully-connected network on each set of hyperparameters and report the total alignment of sorted teachers and students at the end of training as an overall measure of specialization. We fix all other hyperparameters.

When varying second layer learning rate and block length, we fix the regularization strength to $\lambda_{\text{nonneg}} = 0.23$, $\lambda_{\text{norm-L2}} = 0.11$. When varying regularization strength, we fix gate learning rate $\tau_{W^{(2)}}/\tau_{W^{(1)}} = 4$. The regularization strength λ is multiplied separately for each type of regularizer such that $\lambda_{\text{nonneg}} = 10\lambda/11$ and $\lambda_{\text{norm-L2}} = 5\lambda/11$.

B.6 Flexible fully-connected network

We randomly generate two orthogonal teachers $\boldsymbol{W}^{\star m} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. We initialize our fully-connected networks to have two weight layers, $\boldsymbol{W}^{(1)} \in \mathbb{R}^{2d_{\text{out}} \times d_{\text{in}}}$ and $\boldsymbol{W}^{(2)} \in \mathbb{R}^{d_{\text{out}} \times 2d_{\text{out}}}$. We use a faster learning rate and regularize the second layer during training. We treat each weight $W_{ij}^{(2)}$ as a gate that enters into the regularization terms described in Appendix B.3. Student-teacher alignment, total alignment, and gate sorting is then performed as described above.

B.7 MNIST

The original convolutional network features a single convolutional layer with three feature maps and kernel size four, followed by a MaxPool layer of kernel size 2, a ReLU nonlinearity, a and fully-connected sigmoid, ReLU and log softmax layers of size 512, 64, and 10 respectively. The networks are trained using cross-entropy loss with a one-hot encoding for labels. The NTA portion is then trained beginning from the hidden layer representations of the final hidden layer with 64 units. hyperparameters for the NTA portion are given in Table 2 for MNIST and Table 4 for fashionMNIST.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [Yes] We address each of the abstract's claims throughout our manuscript and enumerate our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We end with a section titled conclusions and limitations and mention two broad limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: [Yes] We include derivations for analytically important quantities rather than theorems and proofs. Assumptions in model structure are stated explicitely.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We offer a full description in main text and supplementary material, we also provide a repository containing the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: A public code repository is included.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the relevant sections give details of optimization paradigms used and learning rates. The supplementary contains further experimental details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Where ever appropriate we shaded most lines with SEM bands. Error bars as well when specific points were of interest.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: Code can generally run on any machine type, and was tested on several machine types. Our simulations are rather small and can run in 30 seconds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper presents work whose goal is to advance the field of cognitive science, with possible insights for machine learning.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The algorithm might shed light into cognition but we do not foresee any possible misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: A figure uses a portion of a figure from another work that is under the license "Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND)". This is indicated in the caption.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No research subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.