BE LOUD AND RELEVANT: PRETRAINING OF COMPOSITIONAL TASKS IN LINEAR RNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretraining models with a curriculum of simpler tasks is a common approach to speed up training, but it is unclear what aspects of common task structure benefit learning, or how to choose a theoretically well-justified curriculum. Here we leverage recent learning theory in linear RNNs (Proca et al., 2025), and derive criteria for useful curricula that extend to compositional tasks. We find that pretraining tasks speed up learning and lead to more stereotyped solutions when they train model weights to large magnitude, while also building related structure to the target task. This mirrors results in feedforward networks, but does not rely on previously used statistical arguments of infinite-width networks. Instead we provide a mechanistic description for what constitutes a good CL task in RNNs: we demonstrate that factorized tasks representing the constituent parts of a composed task are ideal curricula because they possess hyperbolic learning dynamics in their pretraining phase that bypass saddles and slow points of the loss landscape.

1 INTRODUCTION

Efficiently training neural network models on complex tasks can be difficult. One approach that often proves useful in practice is pretraining on simpler, related tasks. Curriculum learning (CL), or pretraining more generally, are now ubiquitously used across many domains in machine learning (Soviany et al., 2022; Hacohen & Weinshall, 2019; Narvekar & Stone, 2018), as well as in cognitive and neural science (Ferguson, 1956; Hocker et al., 2025; Behrens et al., 2018; Kepple et al., 2022; Saglietti et al., 2022). Compositional tasks that combine simpler computations to produce more complex behaviors are relevant to all of these fields (Chomsky, 2002; Lake et al., 2016; 2019); however, what makes for a good pretraining task and how to construct effective curricula for compositional tasks remain an open area of study. A key outstanding issue is that we are missing a link between the classical descriptions of CL, which describe its effects as regularizing loss landscapes to avoid suboptimal minima (Bengio et al., 2009), and numerical studies that demonstrate faster speedups. Put simply, we do not yet understand how to choose CL tasks from first principles. Here we make several novel contributions that guide how to choose CL tasks in a principled way:

- We extend the linear RNN learning theory in Proca et al. (2025) to include a broader class of task structure with additional temporal correlations in their task covariances, and derive time to convergence expressions for sequences of such tasks.
- We provide clear principles for which types of tasks improve training speed, and to what accuracy they should be trained to improve overall training speed.
- Our insights generalize beyond linear RNN theory, in particular in compositional tasks.
- We additionally find that while we formulated CL to look for solutions that led to faster overall training, solutions also naturally became more stereotyped.
- To our knowledge, we provide the first mechanistic connection between CL theory and pretraining task choice, demonstrating that good CL tasks naturally avoid saddles and slow points on the loss landscape.

Our approach is as follows: first, we review key aspects of linear RNN learning theory from Proca et al. (2025), and formulate a minimal working example for studying CL (Fig. 1 A): If I can first train on a single CL task before training on a target task, what aspects of their relative task structure

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

will provide a training speedup? We then derive the key inequalities that drive CL learning speed improvement when more general task statistics are possible. Next we explore how these learning speedups manifest at the level of relative task structure between a final target task, and pretraining CL task. We next study how these choices of CL tasks perform in a compositional task. After deriving the geometry of fixed points, we finally study how good CL solutions are positioned relative to saddles on the loss landscape.

Related work

Our results build on a substantial body of work that has studied CL, pretraining, and weight initialization from both statistical and deterministic perspectives, although focused almost entirely on feedforward architectures (Saxe et al., 2014; Chizat et al., 2019). The analysis of learning dynamics for linear RNNs in Proca et al. (2025) provides a key stepping stone to reason about effects of RNN pretraining in precise mathematical terms. Nonetheless, these results are restricted to tasks with no lagged temporal dependencies in the input and input-target covariances. Our work extends linear RNN theory to include broader task structure that might differ across tasks, so that its precise conclusions about training time can apply to tasks trained in sequence.

Studying relative task structure in CL is critical to understanding its utility. In feedforward networks, it had been shown that task misalignment leads to interference towards catastrophic forgetting (Lee et al., 2021). Positive interference between tasks via compositional curricula has been studied in a student-teacher setup (Lee et al., 2024), which revealed a nontrivial interplay between pretraining task complexity and CL efficacy. Such results are typically built on statistical approaches that employ infinitely wide networks, which tends to obscure geometric or mechanistic interpretations for CL, although recent advances describing learning as traversing to saddles along invariant manifolds opens the door to a more nuanced understanding of learning speedups (Zhang et al., 2026).

One way to think about the speedup effects of CL is as setting the initial conditions for the target task optimization. The effects of weight initialization are generally discussed in the domain of lazy vs. rich representation learning. The classic result in feed-forward architectures is that high-variance weights can lead to fast but 'lazy' learning, at the cost of failing to learn 'rich' representations of task structure (Chizat et al., 2019). It's less clear what rich and lazy learning might look like for RNNs. Moreover, the understanding of weight initialization and learning speed has been recently expanding. Braun et al. (2022) demonstrated that feedforward networks can have saddle-free, but rich learning when initial weights are high amplitude. To do so they must have a "zero-balanced" structure that reflects a special orientation of factorized initializations from earlier work in Saxe et al. (2014). The importance of special geometry in initial conditions was also documented empirically in low rank RNNs, but with formal justification still relying on feedforward networks analysis Liu et al. (2024). Bordelon et al. (2025) have studied how slow modes are acquired by gradient-based learning in linear RNNs, but starting from small random initial weights. To our knowledge, a formal analysis of effects of structured weight initialization in RNNs remains an open question.

2 Problem formulation

We will study the effects of task pretraining in linear RNNs of the following form (Fig. 1B):

$$\begin{aligned} \boldsymbol{h}_t &= \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{W}_x \boldsymbol{x}_t \\ \boldsymbol{y}_t &= \boldsymbol{W}_y \boldsymbol{h}_t, \end{aligned} \tag{1}$$

The network state is $\boldsymbol{h}_t \in \mathbb{R}^{N_h \times 1}$, inputs are $\boldsymbol{x}_t \in \mathbb{R}^{N_x \times 1}$, the network's recurrent weight matrix is $\boldsymbol{W}_h \in \mathbb{R}^{N_h \times N_h}$, and the network output is $\boldsymbol{y}_t \in \mathbb{R}^{N_y \times 1}$, so that the output can be expressed as $\boldsymbol{y}_t = \sum_{i=1}^{T} \boldsymbol{W}_y \boldsymbol{W}_h^{T-i} \boldsymbol{W}_x \boldsymbol{x}_i$ (assuming $\boldsymbol{h}_0 = \boldsymbol{0}$). The trainable parameters are $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$, and we hold $\boldsymbol{W}_h$ fixed throughout training. The loss being minimized is the standard mean-squared error loss over a batch of p trials, which for linear RNNs encapsulates a family of tasks that filter (or

integrate) inputs with a bank of linear filters given by $\boldsymbol{W}_h$:

$$\mathcal{L} = \frac{1}{2}\sum_{p}^{P} \|\boldsymbol{y}_p - \hat{\boldsymbol{y}}_{T,p}\|^2.$$

$$= \sum_{t,t'=1}^{T} \frac{1}{2}\,\mathrm{Tr}\left[\boldsymbol{W}_y \boldsymbol{W}_h^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t x_{t'}} \boldsymbol{W}_x^\top \boldsymbol{W}_h^{T-t'\top} \boldsymbol{W}_y^\top - \boldsymbol{W}_y \boldsymbol{W}_h^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t y}\right] + \text{const.} \quad (2)$$

The structure of these linear filtering tasks are fully provided by their task covariances: The input covariance $\boldsymbol{\Sigma}_{x_t x_{t'}}$ is a 4-dimensional tensor that captures how inputs co-vary across input channels and time, while the input-target covariance $\boldsymbol{\Sigma}_{x_t y}$ conveys how inputs and targets are related [1]. They each have a singular value decomposition defined as

$$\boldsymbol{\Sigma}_{x_t x_{t'}} = \mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_{t'}^\top] \approx \sum_{p}^{P} \boldsymbol{x}_{p,t} \boldsymbol{x}_{p,t'}^\top = \boldsymbol{U}_{xx} \boldsymbol{S}_{x_t x_{t'}} \boldsymbol{U}_{xx}^\top$$

$$\boldsymbol{\Sigma}_{x_t y} = \mathbb{E}[\boldsymbol{x}_t \hat{\boldsymbol{y}}_T^\top] \approx \sum_{p}^{P} \boldsymbol{x}_{p,t} \hat{\boldsymbol{y}}_{p,T}^\top = \boldsymbol{U}_{xy} \boldsymbol{S}_{x_t y} \boldsymbol{V}_{xy}^\top. \quad (3)$$

Our general goal here is to determine the time τ that it takes a network to learn a target task "2", and to contrast that learning with a scenario where the network starts by training on another task "1", then switches to the target. In particular, we want to understand under what circumstances and for what kind of pairs of tasks (each with a similar geometric alignment of their input-input and input-target covariance functions) that training the sequence offers i) speed benefits relative to training the target alone $\tau_{0-2} > \tau_{0-1} + \tau_{1-2}$, as well as providing more stereotyped solutions (Fig. 1A).

We will consider two families of target tasks in this work. The first form is a "factorized" task in which rigorous theoretical analysis about learning dynamics has been derived (Proca et al., 2025) (Fig. 1C). Factorized tasks have a strict relationship between $\boldsymbol{\Sigma}_{x_t x_{t'}}$ and $\boldsymbol{\Sigma}_{x_t y}$ that impart trained networks with distinct input channels to contribute to individual outputs of the network (when rotated into the correct reference frame $\tilde{x}, \tilde{y}$). If networks are initialized such that these "modes" do not overlap, saddle-free learning dynamics are guaranteed (Proca et al., 2025). The other case we consider are a minimal example of functionally compositional tasks, which require combining information from across factorized modes to perform a target computation (Fig. 1D). These compositional tasks do not fall in the scope of the previous linear RNN theory, but here we study the degree to which its insights will hold in compositional tasks. For those comparisons (Sec. 4),we will consider canonical 2D examples of a factorized and compositional task:

$$W_y = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}, \qquad W_x = \mathbb{I}, W_H = \Lambda \qquad \text{(factorized task)} \quad (4)$$

$$W_y = \begin{bmatrix} C & 0 \\ C & C \end{bmatrix}, \qquad W_x = \mathbb{I}, W_H = \Lambda \qquad \text{(compositional task)} \quad (5)$$

These examples possess diagonal recurrence matrices, and output weights that either scale (factorized), or couple and scale (compositional) network-filtered inputs by a constant factor C.

3 Curriculum learning principles derived from factorized tasks

A primary objective of curriculum learning is to speed up training, and our starting point is a result from Proca et al. (2025), which derives a closed-form expression for the hyperbolic learning dynamics of factorized tasks under special initial conditions. In this work we extend upon their formulation to allow for non-whitened input correlations and nontrivial temporal input-target correlations, which allows us to analyze the training time of different tasks in sequence. This derivation is detailed in Appendix A.1 in order to focus on the key result of training time speedup.

[1] We refer to the transpose $\boldsymbol{\Sigma}_{x_t y}$ for a fixed time point via its indices as $\boldsymbol{\Sigma}_{x_t y}^\top = \boldsymbol{\Sigma}_{y x_t}$.

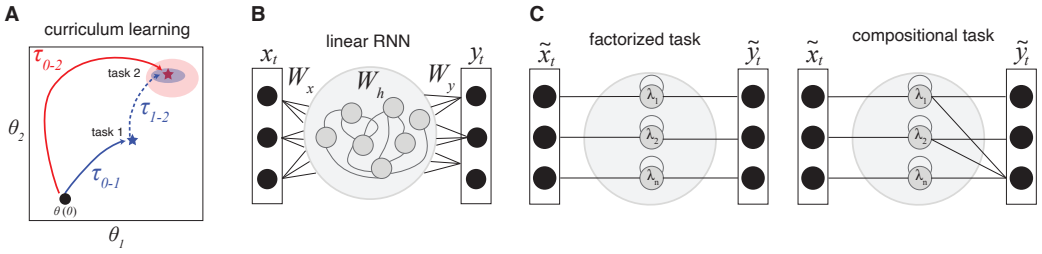

Figure 1: **A.** The goals of curriculum learning. The time τ needed to learn a task $\mathcal{T}_2$ can be potentially accelerated by first learning a different task $\mathcal{T}_1$. The potential speedup $\tau_{0-2} - (\tau_{0-1} + \tau_{1-2})$ will depend upon the similarity between the tasks' correlational structures. Curriculum learning can also lead to cmore stereotyped solutions with lower variance in parameters across seeds (colored ovals) **B.** Linear RNN architecture. Only W_x and W_y are trained, and W_h are fixed, to facilitate closed-form solutions for their training dynamics. **C.** Primary task types studied with linear RNNs here. (left) Factorized tasks process individual inputs through distinct network modes, and couple to individual output channels, allowing for closed-form analysis of learning dynamics. (right) Compositional tasks combine computations from multiple factorized modes to perform complex operations.

Saddle-free learning of factorized tasks is possible if the initial conditions for weight matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ lie along the same distinct, orthogonal directions. Specifically, this requires that the columns of $\boldsymbol{W}_{x_{:,\alpha}}(0) = a_\alpha \boldsymbol{r}_\alpha$ and the rows of $\boldsymbol{W}_{y_{\alpha,:}}(0) = c_\alpha \boldsymbol{r}_\alpha$, for some orthogonal basis set $\{\boldsymbol{r}\} \in \mathbb{R}^{N_h \times 1}$. Intuitively, this means that input and output weights should be initialized onto the same set of non-overlapping factorized modes. Initialized this way, the product of those coefficients for each mode, $u_\alpha = a_\alpha c_\alpha$ follows a saddle-free, sigmoidal learning trajectory over optimization time τ,

$$a_\alpha(\tau)c_\alpha(\tau) = \frac{1}{\left[\frac{1}{(a_\alpha(0)c_\alpha(0))} - \frac{\beta^\alpha_{xx}}{\beta^\alpha_{yx}}\right] e^{-2\tau\beta^\alpha_{yx}/\gamma} + \frac{\beta^\alpha_{xx}}{\beta^\alpha_{yx}}}. \tag{6}$$

γ is a learning rate, and β^α_{yx},β^α_{xx} reflect the recurrence-weighted singular values of task covariances:

$$\beta^\alpha_{yx} = \sum_t^T \lambda_\alpha^{(T-t)} s^\alpha_{yx_t}, \qquad \beta^\alpha_{xx} = \sum_{t,t'}^T \lambda_\alpha^{(2T-t-t')} s^\alpha_{x_t x_{t'}}. \tag{7}$$

λ are the eigenvalues of the recurrence matrix, and s_{y,x_t}, s_{x_t,x'_t} are the singular values of the task covariances. Due to the non-overlapping initialization, the coefficients for each mode α optimize independently of the others,can be performed in parallel, and depend purely upon which task covariance singular values and recurrence eigenvalues it uses. Note that there is a degeneracy in solutions for factorized tasks that allows rescaling of the form $ac = (ad)(d^{-1}c)$ for a scalar d, which at the matrix level reflects scaling invariance of outputs by a diagonal matrix D: $y_t = \sum_t W_y D \Lambda D^{-1} W_x x_t$.

We now consider training two factorized tasks in sequence, defined by their task covariances as $\mathcal{T}_k = \{\boldsymbol{\Sigma}^{(k)}_{xx}, \boldsymbol{\Sigma}^{(k)}_{xy}\}, k = 1, 2$. Optimal solutions u^* to factorized tasks occur when $\tau \to \infty$:

$$u^{*(k)} = \beta^{(k)}_{yx} / \beta^{(k)}_{xx} = \frac{\sum_i^T b^{T-t} s^{(k)}_{yx_t}}{\sum_{t,t'}^T b^{2T-t-t'} s^{(k)}_{x_t x_{t'}}}. \tag{8}$$

For a mode to be ϵ close to this solution, $u(\tau) = (1-\epsilon)u^{*(1)}$. The training time to reach $u(\tau)$ is

$$t_{i\to 1} = \frac{\gamma}{2\beta^{(1)}_{yx}} \left(\log \left| \frac{u^{*(1)}}{u_0} - 1 \right| - \log \left| \frac{\epsilon}{1-\epsilon} \right| \right). \tag{9}$$

Our primary result determines the conditions under which training on $\mathcal{T}_1$ offers a speedup when learning a task $\mathcal{T}_2$ with equivalent task geometry (derivation in Appendix A.2),

$$t_{i\to 2} > t_{i\to 1} + t_{1\to 2}$$

$$\log\left|\frac{u^{*(2)}}{u_0}-1\right|+\frac{\beta_{yx}^{(2)}}{\beta_{yx}^{(1)}}\log\left|\left(\frac{\epsilon^{(1)}}{1-\epsilon^{(1)}}\right)\left(\frac{u_0}{u^{*(1)}-u_0}\right)\right|-\log\left|\frac{u^{*(2)}}{(1-\epsilon)u^{*(1)}}-1\right|>0 \quad (10)$$

Eq. 10 is complex, and drawing insights from satisfying all of conditions simultaneously is difficult. Instead, we focus on each term individually to see what it conveys about curriculum learning speedups. The first term shows that starting from small initial conditions relative to the 2nd task target provides a speedup. The other terms reveal key principles for CL tasks to speed up training:

1. $\beta_{yx}^{(1)} > \beta_{yx}^{(2)}$ (2nd term, 3rd term rewritten in Appendix A.2). This implies task 1 should have larger, more salient input-target coupling, which is in line with it being an "easier" task. It also predicts that the first task should have larger amplitude parameters than task 2. [2]
2. $\beta_{xx}^{(1)} < \beta_{xx}^{(2)}$ (3rd term, rewritten in Appendix A.2). This implies that initial tasks should have less variance in their inputs and be more predictable, again a trait of easier tasks. Note that this relationship assumes $\beta_{yx}^{(1)} = \beta_{yx}^{(2)}$.
3. Initial tasks do not need to be trained to high fidelity in order to speed up learning (3rd term). The singularity in the 3rd term occurs when $\mathcal{T}_1$ has been optimized to exactly be the solution to task $\mathcal{T}_2$, which produce a CL speedup when it is in the neighborhood of this singularity. Depending on the magnitude of the two optimal solutions, this speedup can occur for small accuracy on the first task.

Importantly, Eq. 10 only holds if the geometry of $\mathcal{T}_1$ is structurally equivalent to $\mathcal{T}_2$, meaning that the the SVD eigenvectors for the task covariances are the same in both tasks. Otherwise, after training on $\mathcal{T}_1$, the initial conditions would not lie in an orthogonal basis set by the eigenvectors of $\mathcal{T}_2$, and there would be cross-mode contributions during training (numerical results in Appendix A.6.2).

We numerically validated that our insights from Eq. 10 hold. First, we focused on modulating singular value magnitudes while omitting temporal correlation; we found that CL tasks with larger Σ_{yx_t} singular values led to faster training (Fig. 2**A**-**C**), and conversely that tasks with smaller $\Sigma_{x_t,x_t'}$ singular values led to faster training (following principles 1 and 2). In practice this means that tasks that tune input and output weights to initially larger values aid in learning later tasks with smaller weights. We additionally noted the predicted presence of the singularity in task training time in both scenarios (principle 3, Figs. 2**A**,**D**, cyan line), which created a support of faster overall training when the first task was trained to suboptimal accuracy. To study temporal correlations in the inputs, we generated temporally correlated inputs with correlation strength K (Fig. 2F), and to study temporal structure in Σ_{yx_t} we studied leaky integrator tasks with a temporal decay constant κ (Fig. 2G, detail provided in Appendix A.6.1). We found that stronger correlations K in the inputs of $\mathcal{T}_1$ improve training speed (principle 2, Fig. 2E), and that CL tasks with longer integration windows lead to faster training on $\mathcal{T}_2$ (principle 1, Fig. 2H). We also found evidence of training time singularity when modulating κ (principle 3). Inputs with more temporal correlation are intuitively "easier" because larger input correlation strength leads to a more stereotyped and predictable signal. It is less intuitive how longer integration windows would constitute an easier task, and we view this as interesting result where learning theory can make predictions for good CL tasks that might defy natural intuition. We confirm that these results hold for nonlinear RNNs, joint training of recurrent weights, and that the results collapse to feedforward results without recurrence (Appendices A.6.3-A.6.5). Additional numerical comparisons to theory are provided in Appendix A.6.6.

4 Compositional tasks

Having established theoretical CL principles in factorized tasks, we next asked to what extent these principles hold for a compositional task. Here we used Eq. 5 as the target for $C = 2.0$, and kept $\boldsymbol{W}_x = \mathbb{I}$. We studied several potential CL tasks: i) the related factorized task ("factorized" Eq. 4), ii) a scaled version of the target task with a different C value ("scaled target"), iii) permutations of the target task that swapped entries of W_y ("task permutations"), iv) low-rank versions of the task that kept individual columns of $\boldsymbol{W}_y$ and set the other target entries to 0, and finally v) random tasks where $\boldsymbol{W}_{y_{i,j}} \sim \mathcal{N}(C, 1)$. Explicit descriptions all CL tasks are provided in Appendix A.5.

[2] This is in contrast to what the first term Eq. 10 suggests; however multiple terms suggesting that larger initial weights are beneficial, likely drown out this contribution.

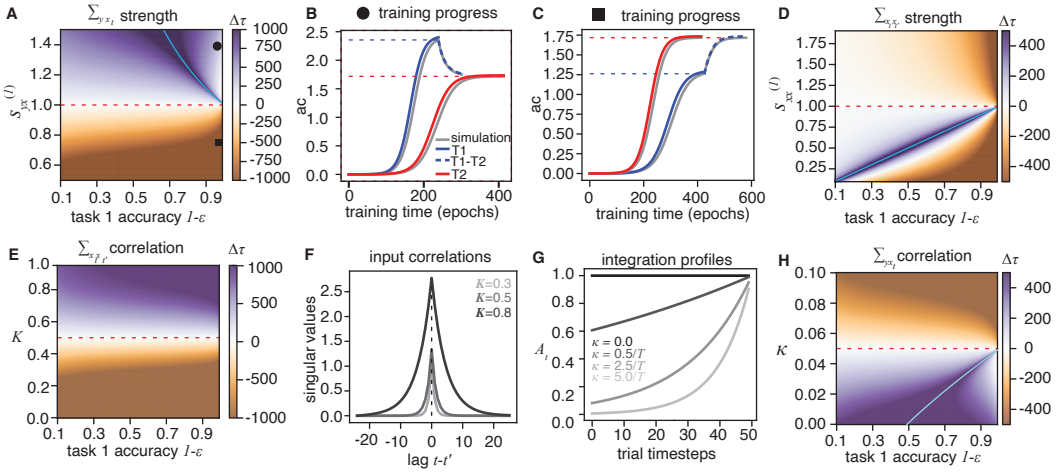

Figure 2: Effect of CL task covariance magnitude (**A-D**) and temporal correlation (**E-H**) on training speed. **A.** Phase portrait of difference in training time between direct $\mathcal{T}_2$ training and a curriculum of $\mathcal{T}_1$ (to accuracy $1-\epsilon$) followed by task $\mathcal{T}_2$. $\mathcal{T}_1$ accuracy and singular value strength or $\mathbf{\Sigma}_{yx_t}$ were modulated. Dotted red line shows singular value of $\mathcal{T}_2$. Cyan denotes singularity when $\mathcal{T}_1$ solution corresponds to $\mathcal{T}_2$ optimum, requiring no training time. **B-C.** Predicted vs. numerical optimization training trajectories for individual parameter settings, denoted by square and circle in panel **A**. **D.** Similar phase portraits as in **A**, but for modulating singular value of $\mathbf{\Sigma}_{x_t,x'_t}$. **E.** same as in **D**, but for temporal corelation strength K in inputs following an AR1 process in **F**. **G** Modulating temporal correlations of Σ_{yx_t} through changing integration window of a leaky integrator. **H** Same as in **A**, but for temporal integration time constant κ.

Here we focus on one main principle from factorized learning theory, which is that CL tasks that train to larger weights can offer a training speedup, and consider how scaled versions of each of these CL tasks perform by modulating the scaling parameter C. Additionally, we consider whether alignment of task structure is an important component for good CL performance. Factorized and scaled-target tasks possess closely-related structure to the target task, while task permutations match the target in their overall rank and magnitude, but lack the true structure of the target. Low-rank tasks are rank-deficient, but share some of the target structure, while random tasks act as our null model for task structure comparisons.

4.1 Good CL tasks have structurally related large magnitude weights

We compared the training time for sequential training of the CL task followed by the target task to direct target task training (Fig. 3). We found that the CL strategies had slower training when the magnitude of $\boldsymbol{W}_y$ for task 1 was weaker than the target task. As its magnitude increased to larger values, CL tasks with closely related structure began to outperform direct training (Fig. 3**A**). In particular, factorized analogs of the target task showed continuing improvement in training speed as the magnitude of its CL task target increased. Scaled versions of the target task as CL also had a subtle speedup at intermediate magnitudes. CL that used target permutations or rank-deficient analogs to the target saturated at negligible or only marginally better speedups (see Appendix A.6.7).

We also noted that the variance in training times across seeds for many of the CL strategies was substantially reduced relative to direct training, which we reasoned to potentially reflect more stereotyped solutions. To asses this, we rotated all of the resulting optimal solutions into a standard coordinate frame that accounts for the degeneracy of solutions via $\tilde{\boldsymbol{W}}_y = \boldsymbol{W}_y \boldsymbol{W}_x^{-1}$, and then looked at the spread of solutions in PC space (Fig. 3**B**). CL tasks using larger magnitude targets led to more stereotyped solutions with a smaller variance in parameters. By tracking the norm of the covariance matrix, we found that similar stereotyped solutions occur as the magnitude of the CL pretraining task becomes larger than the target task (Fig. 3**C**). In summary, pretraining tasks with larger magnitude than a target task that also possess related task structure, in particular factorized variants that train on individual elements of a composed task, lead to faster training and more stereotyped solutions.

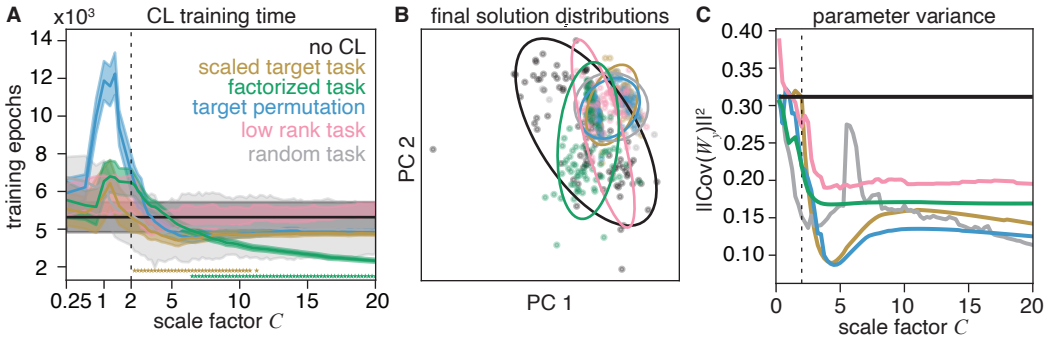

Figure 3: **A.** Total training time required for compositional tasks using curriculum learning. Scale factors less than $C = 2.0$ are on a log scale, ($* : p < 0.001$, sign-rank test) **B.** Distribution of optimized weights for $C = 15.0$. Ellipses show 2.0 std. **C.** Variance in task solutions as a function of the scaling of output weight parameter C. For visual clarity we only show a single representative task permutation and low-rank task in plots, but full results are provided in Appendix A.6.7.

4.2 PRETRAINING TASKS OVERSHOOT AND AVOID SADDLES ON THE LOSS LANDSCAPE

A key aspect impacting training speed are the number and location of saddles. When examining the training progress of each CL type, we found that direct training encountered saddles that were characterized by plateaus in training loss and temporary dips and rises in the norm of the gradient, but that CL solutions trained on large magnitude tasks did not (Fig. 4 **A**-**C**). Since our intuitions from factorized learning theory seemed to hold in compositional tasks, we reasoned that an analysis of saddles for factorized tasks could provide mechanistic insight. A comprehensive discussion is provided in Appendix A.3, and we provide the primary description of saddles here, which is based on Baldi & Hornik (1989). The gradients of the loss are

$$\frac{\partial \boldsymbol{W}_x}{\partial \tau} = \sum_{t,t'} \boldsymbol{W}_h^{T-t\top} \boldsymbol{W}_y^\top \left(\boldsymbol{\Sigma}_{yx_t} - \boldsymbol{W}_y \boldsymbol{W}_h^{T-t'} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t x_{t'}} \right)$$
$$\frac{\partial \boldsymbol{W}_y}{\partial \tau} = \sum_{t,t'} \left(\boldsymbol{\Sigma}_{yx_t} - \boldsymbol{W}_y \boldsymbol{W}_h^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t x_{t'}} \right) \boldsymbol{W}_x^\top \boldsymbol{W}_h^{T-t'\top}. \tag{11}$$

Fixed points occur when $\frac{\partial \boldsymbol{W}_x}{\partial \tau} = \frac{\partial \boldsymbol{W}_y}{\partial \tau} = 0$, which in the above equations occur when the network aligns with modes of the input-target covariance. The global minimum occurs when the RNN aligns with all modes of $\boldsymbol{\Sigma}_{yx_t}$, saddles occur when network only aligns with subset of modes, and a local maximum of exists when the RNN aligns with no network modes (the trivial $\boldsymbol{W}_x = \boldsymbol{W}_y = \boldsymbol{0}$ solution). For a network with N factorized modes this leads to $2^N - 2$ saddles. In our case we assume fully uncorrelated white noise ($\boldsymbol{\Sigma}_{x_t x_{t'}} = \mathbf{I}$), and the saddles occur at

$$\boldsymbol{W}_y^{s1} = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} D, \quad \boldsymbol{W}_x^{s1} = D^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$\boldsymbol{W}_y^{s2} = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} D, \quad \boldsymbol{W}_x^{s2} = D^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{12}$$

where D is a diagonal matrix. D accounts for the previously mentioned invariance in network activity for relative weight differences between $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$, which here defines the *manifold* of saddles present for factorized tasks. The loss function on each saddle manifold will reflect which singular values of $\boldsymbol{\Sigma}_{yx_t}$ were captured by the network, with smaller losses occurring when larger singular values are well-represented by the network (see Appendix A.3).

We next visualized how initial parameters, optimal solutions and saddles were organized in PC space for factorized tasks (Fig. 4 **D**). Specifically, we used PCA on the 8-D weight space, then visualized numerical results in the top 2 PCs. We then discretized this 2D space and calculated the loss at these points to determine where optima and saddles exist. Additionally, we visualized the theoretical saddle manifold by bootstrapping the saddles in Eq. 12, using a range of D matrices. We found

that initial conditions were oriented such that sometimes saddles would be encountered, but at times saddles could be avoided. The organization of the optima, minima, and saddle manifolds shows that saddles are more likely to be encountered for smaller weights lying closer to the origin, but that cones of saddle-free regions exist at weights larger than the optimal task solution. [3]

We next performed the same analysis on our compositional task, and visualized where CL pretraining places solutions relative to the target task optima. We found that there were subtle changes in the fixed point structure when comparing compositional tasks to factorized ones: the originally distinct optima relaxed into a wider basins, but still with 4 distinct regions. We also examined the norm of the gradient in this space and found that the original fixed point locations shifted into larger regions of slow points (Appendix A.6.8). However, the general loss landscape structure appeared to stay intact. When we visualized the location of weights after CL training we found strategies using large-magnitude weights and related structure do indeed "overshoot" the saddles and slow points encountered by direct training. For the same scaling factor size C, factorized CL placed solutions closer to the final solution than full-rank task permutation CL, presumably allowing for faster target training. Low rank solutions placed weights near "ghost points" (Sussillo & Barak, 2013), the locations where factorized saddles were originally located, which slowed optimization. Here the geometry of the loss landscape explains why low-rank solutions trained slowly, since its fixed point analysis demonstrated that saddles occur in rank-deficient solutions.

In summary, we found that fast CL strategies allow for saddle-free learning of the target task by effectively avoiding saddles and slow regions on the loss landscape. Avoiding these saddles occurs by using larger magnitude weights in tasks with relevant structure, including the same rank as the target task. Factorized tasks comprising the primitive computational modes of the target task best balanced the requirement to overshoot saddles alongside staying close to the final task solution.

5 Discussion

Our work set out to provide a theoretical basis for choosing good curriculum learning tasks in linear RNNs, and to test whether such a theory would inform more complicated compositional tasks. Building upon recent theoretical results in Proca et al. (2025), we derived how the strength and temporal structure of task covariances drives pretraining efficiency when tuning the input and output weights of a linear RNN. We identified three primary insights about CL in this regime: i) Pretraining to larger weights than the target speeds up learning, ii) more stereotyped and predictable input signals aid in learning, and iii) pretraining does not need to converge to high accuracy to be beneficial. We demonstrated that compositional tasks, which fall outside the formal scope of our theory, benefited from the first principle. Specifically, we found that CL tasks that trained weights to relatively large amplitudes aided in learning, but that they needed to possess structure that was related to the task. Factorized tasks that encompass all of the primitive computations that went into the target performed particularly well, while rank-deficient tasks and other forms that lacked task structure did not. We demonstrated that the reason behind this CL performance was that training to large weights systematically avoided saddles in the loss landscape, and possessing related structure placed solutions closer to the final target (and further away from slow "ghost" points supported by former saddles of the factorized task) to allow for saddle-free optimization.

In practice, intuitively easier tasks are chosen for curricula, and this idea has motivated much of the successful use of curriculum learning (Soviany et al., 2022). The principles we described show specific task dimensions where increasing difficultly is important, and we can envision recommendations for good curricula in different research domains where RNNs are typically used. In behavioral sciences for example, animals typically undergo intensive shaping procedures (Pinto et al., 2018; International Brain Laboratory et al., 2021; Mah et al., 2023; Brunton et al., 2013). Our result that stronger input-target coupling makes for good initial tasks suggests shaping with the strongest task stimuli first, and then moving to weaker ones. Or conversely, shaping with the largest rewards before introducing smaller ones. These suggestions dramatically over-simplify other processes that might be steering behavior in living systems, but they act as potential guidance for choosing tasks.

[3] We note some small deviations from numerical optima from simulations with the phase portrait in 4 **D-E**). This is an unavoidable consequence of the low dimensional projection, and relevant structure likely exists beyond what we can show here. Similar issues are present when bootstrapping fixed points via D.

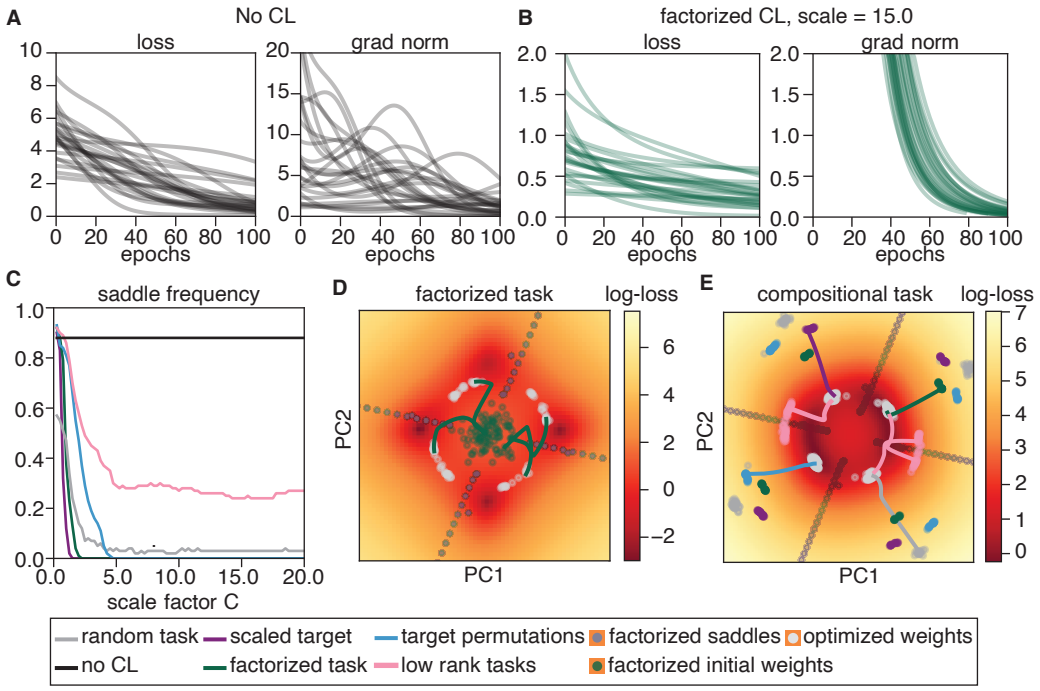

Figure 4: **A.** Loss and gradient for direct target task training. **B.** Same for factorized CL training to large magnitude. **C.** Frequency of encountering saddles across $N = 100$ seeds. **D.** Phase portrait of loss landscape for a factorized task. Saddle locations are derived analytically (see main text). **E.** Phase portrait for compositional task, showing similar structure as factorized task, but with larger spread of slow points rather than isolated fixed points. Factorized saddles are also shown.

Our approach to mechanistically understand CL performance in compositional tasks is distinct, in that it makes a connection to a core CL theory of avoiding suboptimal fixed points. We showed that saddles define boundaries in weight space, and that larger weights offer a larger region of starting conditions to avoid saddles, which would be interesting to study in higher dimensional tasks. Such tasks could couple different modes and would require deeper characterization of fixed points to understand their geometry, which is beyond the scope of this work. However, it offers the possibility of understanding how to sequence multiple pretraining tasks.

The training time singularity also suggests a potential heuristic for when to stop training on a pretraining task in model simulations. Factorized tasks have hyperbolic learning, and tracking the loss of the target task for rapid changes while optimizing the pretraining task could act as a useful diagnostic for when to stop training. Observing a rapid change in the target loss suggests that training has placed you beyond its inflection point (Eq. 8), and in the neighborhood of the target solution.

A limitation of our current results is that the theory does not easily extend to jointly training recurrent weights, though we did observe numerically that some of the core principles hold in this regime as well (Appendix A.6.5). Our future work aims to understand the contributions of recurrence in CL from a geometric and mechanistic perspective, from which we can draw from a rich body of work understanding how dynamical systems motifs are organized and learned when performing multiple tasks (Turner & Barak, 2023).

Reproducibility Statement

In order to facilitate reproducibility of our results we have included a full derivation of the theory in the Appendix. We have also provided details for the simulation and training of RNNs that lead to the results. All code for generating the results in the manuscript will be provided at the time of publication.

AI USE DISCLOSURE

In this work we used generative AI (Claude) to aid in managing the batching of the various curricula studied here. Specifically, we used it to copy the structure of existing CL code and to rename files and to make minor hyperparameter changes. We also used it to analyze the clarity of some core training code, and to provide suggestions for making it more efficient. We also used generative AI to aid in theorem development by asking it to analyze core expressions in our work. At no time did we use generative AI to write any text or theorems, nor did we use it to modify any figures. We have reviewed all AI-assisted work, and we take responsibility for any claims or artifacts produced with the aid of generative AI.

REFERENCES

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Blake Bordelon, Jordan Cotler, Cengiz Pehlevan, and Jacob A Zavatone-Veth. Dynamics of learning to integrate in linear recurrent neural networks. *arXiv preprint arXiv:2503.18754*, 2025.

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.

Bingni W Brunton, Matthew M Botvinick, and Carlos D Brody. Rats and humans can optimally accumulate evidence for decision-making. *Science*, 340(6128):95–98, 2013.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Noam Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

George A Ferguson. On transfer and the abilities of man. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 10(3):121, 1956.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pp. 2535–2544. PMLR, 2019.

David Hocker, Christine M Constantinople, and Cristina Savin. Compositional pretraining improves computational efficiency and matches animal behaviour on complex tasks. *Nature Machine Intelligence*, pp. 1–14, 2025.

IBL International Brain Laboratory, Valeria Aguillon-Rodriguez, Dora Angelaki, Hannah Bayer, Niccolo Bonacchi, Matteo Carandini, Fanny Cazettes, Gaelle Chapuis, Anne K Churchland, Yang Dan, et al. Standardized and reproducible measurement of decision-making in mice. *Elife*, 10: e63711, 2021.

Daniel R Kepple, Rainer Engelken, and Kanaka Rajan. Curriculum learning as a tool to uncover learning principles in the brain. In *International Conference on Learning Representations*, 2022.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *arXiv preprint arXiv:1604.00289*, 2016.

Brenden M Lake, Tal Linzen, and Marco Baroni. Human few-shot learning of compositional instructions. *arXiv preprint arXiv:1901.04587*, 2019.

JH Lee, SS Mannelli, and A Saxe. Why do animals need shaping? a theory of task composition and curriculum learning, 2024.

Sebastian Lee, Sebastian Goldt, and Andrew Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pp. 6109–6119. PMLR, 2021.

Yuhan Helena Liu, Aristide Baratin, Jonathan Cornford, Stefan Mihalas, Eric Shea-Brown, and Guillaume Lajoie. How connectivity structure shapes rich and lazy learning in neural circuits, 2024. URL https://arxiv.org/abs/2310.08513.

Andrew Mah, Shannon S Schiereck, Veronica Bossio, and Christine M Constantinople. Distinct value computations support rapid sequential decisions. *Nature communications*, 14(1):7573, 2023.

Sanmit Narvekar and Peter Stone. Learning curriculum policies for reinforcement learning. *arXiv preprint arXiv:1812.00285*, 2018.

Lucas Pinto, Sue A Koay, Ben Engelhard, Alice M Yoon, Ben Deverett, Stephan Y Thiberge, Ilana B Witten, David W Tank, and Carlos D Brody. An accumulation-of-evidence task using visual pulses for mice navigating in virtual reality. *Frontiers in behavioral neuroscience*, 12:36, 2018.

Alexandra Maria Proca, Clémentine Carla Juliette Dominé, Murray Shanahan, and Pedro A. M. Mediano. Learning dynamics in linear recurrent neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KGOcrIWYnx.

Luca Saglietti, Stefano Mannelli, and Andrew Saxe. An analytical theory of curriculum learning in teacher-student networks. *Advances in Neural Information Processing Systems*, 35:21113–21127, 2022.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *The Second International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6120.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

Elia Turner and Omri Barak. The simplicity bias in multi-task rnns: shared attractors, reuse of dynamics, and geometric representation. *Advances in Neural Information Processing Systems*, 36:25495–25507, 2023.

Yedi Zhang, Andrew Saxe, and Peter Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *International Conference on Learning Representations*, volume 2026, pp. 57507–57551, 2026.

A Appendix

A.1 Extended derivation of learning dynamics

In this section we derive Eq. 6. Our approach is based on Proca et al. (2025) and Saxe et al. (2014), and considers a broader range of tasks with fewer constraints on the task geometry. While we ultimately consider a regime that is similar to Proca et al. (2025), we aim to keep the derivation as general as possible and highlight when assumptions are required to yield tractable analytical solutions. We hope that this exposes future directions for the theory of learning in RNNS.

We begin with a linear RNN of the form

$$\boldsymbol{h}_t = \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{W}_x \boldsymbol{x}_t \tag{13}$$

$$\boldsymbol{y}_t = \boldsymbol{W}_y \boldsymbol{h}_t, \tag{14}$$

which maps time-varying inputs $\boldsymbol{x}_t \in \mathbb{R}^{N_x \times 1}$ into a network state $\boldsymbol{h}_t \in \mathbb{R}^{N_h \times 1}$, read out into outputs $\boldsymbol{y}_t \in \mathbb{R}^{N_y \times 1}$. The learnable parameters of the network include the recurrent weight matrix $\boldsymbol{W}_h \in \mathbb{R}^{N_h \times N_h}$, input matrix $\boldsymbol{W}_x \in \mathbb{R}^{N_h \times N_x}$, and output matrix $\boldsymbol{W}_y \in \mathbb{R}^{N_y \times N_h}$. The RNN will be optimized to perform on task pulled from a family of leaky integration tasks, where inputs $\boldsymbol{x}_{1:T}$ are integrated over time with different linear filters to yield outputs, $\boldsymbol{y}_T$, at the end of the trial, T. The loss over a batch of P trials for this single output scenario of generating a target $\hat{\boldsymbol{y}}$ is given as

$$\mathcal{L} = \frac{1}{2} \sum_{p}^{P} \|\boldsymbol{y}_p - \hat{\boldsymbol{y}}_{T,p}\|^2. \tag{15}$$

Starting from initial state $\boldsymbol{h}_0 = \boldsymbol{0}$, the network dynamics evolve as

$$\boldsymbol{h}_t = \sum_{i=1}^{t} \boldsymbol{W}_h^{t-i} \boldsymbol{W}_x \boldsymbol{x}_i, \tag{16}$$

which allows the loss to be rewritten as

$$\mathcal{L} = \sum_{t,t'=1}^{T} \frac{1}{2} \operatorname{Tr}\left[\boldsymbol{W}_y \boldsymbol{W}_h^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t x_{t'}} \boldsymbol{W}_x^\top \boldsymbol{W}_h^{T-t'\top} \boldsymbol{W}_y^\top - \boldsymbol{W}_y \boldsymbol{W}_h^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t y} \right] + \text{const.} \tag{17}$$

The autocorrelation function of the inputs, $\boldsymbol{\Sigma}_{x_t x'_t} = \mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_{t'}^\top]$, together with the cross-correlation between time-varying input and targets, $\boldsymbol{\Sigma}_{x_t y} = \mathbb{E}[\boldsymbol{x}_t \hat{\boldsymbol{y}}_T^\top]$ fully specify an instance of the task. Different tasks will have different $\boldsymbol{\Sigma}_{x_t x'_t}$ and $\boldsymbol{\Sigma}_{x_t y}$, with varying degrees of overlap.

The loss in Eq. 17 depends on the learnable parameters, as well as data-averaged task covariances $\Sigma_{x_t x_{t'}}$ that describe how inputs co-vary over time and input dimensions, as well as input-target covariances $\Sigma_{x_t y}$ that describe how inputs co-vary with target values. These covariances have singular value decompositions (SVD) given by

$$\boldsymbol{\Sigma}_{x_t x_{t'}} = \mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_{t'}^\top] \approx \sum_{p}^{P} \boldsymbol{x}_{p,t} \boldsymbol{x}_{p,t'}^\top = \boldsymbol{U}_{xx} \boldsymbol{S}_{x_t x_{t'}} \boldsymbol{U}_{xx}^\top \tag{18}$$

$$\boldsymbol{\Sigma}_{x_t y} = \mathbb{E}[\boldsymbol{x}_t \hat{\boldsymbol{y}}_T^\top] \approx \sum_{p}^{P} \boldsymbol{x}_{p,t} \hat{\boldsymbol{y}}_{p,T}^\top = \boldsymbol{U}_{xy} \boldsymbol{S}_{x_t y} \boldsymbol{V}_{xy}^\top \tag{19}$$

Consistent with the previous work (Proca et al., 2025), we make additional assumption regarding the form of the task covariances: 1) we assume a static "task geometry," meaning that that the SVD axes ($\boldsymbol{U}_{xx}, \boldsymbol{U}_{xy}, \boldsymbol{V}_{xy}$) are constant over time, which implies a constant input-to-output mapping during the task. We note that this is an assumption that is violated for compositional tasks. Unlike previous work, we do not assume fully whitened inputs here. When considering learning a sequence of tasks this assumption would be too restrictive: while it is possible to fully whiten a target task, the corresponding coordinate system will not necessarily whiten inputs for the pretraining tasks.

Next, we recast the loss function in a rotated space that couples singular values of $\boldsymbol{\Sigma}_{x_t y}$ with recurrent modes provided by a Schur decomposition as $\boldsymbol{W}_h = \boldsymbol{U}_h \boldsymbol{H}_h \boldsymbol{U}_h^\top$, where $\boldsymbol{H}_h$ is upper-triangular for non-normal dynamics, and diagonal for normal dynamics. Rotating the input and output weights as $\boldsymbol{W}_x = \boldsymbol{U}_h \tilde{\boldsymbol{W}}_x \boldsymbol{U}_{xx}^\top$, $\boldsymbol{W}_y = \boldsymbol{V}_{xy} \tilde{\boldsymbol{W}}_y \boldsymbol{U}_h^\top$, the loss function becomes

$$\mathcal{L} = \sum_{t,t'}^{T} \frac{1}{2} \operatorname{Tr}\left[\tilde{\boldsymbol{W}}_y \boldsymbol{H}_h^{T-t} \tilde{\boldsymbol{W}}_x \boldsymbol{S}_{x_t x_{t'}} \tilde{\boldsymbol{W}}_x^\top \boldsymbol{H}_h^{T-t'\top} \tilde{\boldsymbol{W}}_y^\top \right] - \operatorname{Tr}\left[\tilde{\boldsymbol{W}}_y \boldsymbol{H}_h^{T-t} \tilde{\boldsymbol{W}}_x (\boldsymbol{U}_{xx}^\top \boldsymbol{U}_{xy}) \boldsymbol{S}_{x_t y} \right]. \tag{20}$$

We make a further assumption here that that $\boldsymbol{U}_{xx}^\top \boldsymbol{U}_{xy} = \boldsymbol{I}$, which holds only for $\boldsymbol{U}_{xx} = \boldsymbol{U}_{xy}$. This implies a connection between the axes of the input covariability and the SVD modes of the input-to-output mapping, which is that the directions of variability in the inputs must be aligned

with the primary SVD modes of $\mathbf{\Sigma}_{x_t y}$. [4]. We note that this is another assumption that is violated for compositional tasks. At this stage, we do not assume that input and output matrices are naturally aligned to the network and singular values modes, meaning $\tilde{\boldsymbol{W}}_x$ and $\tilde{\boldsymbol{W}}_y$ are not assumed to be diagonal. We will implement this in practice when by choosing a privileged set of initial conditions for training, but our derivation does not require this. This means that factorized modes may, in principle, utilize multiple modes of the network.

We restrict the learning dynamics to how input and output parameters to the network update over learning, as the learning dynamics for recurrent weights do not have analytical solutions without introducing approximations. By denoting the learning trajectory by a variable τ, these updates are given by

$$\frac{\partial \tilde{\boldsymbol{W}}_x}{\partial \tau} = -\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{W}}_x} = \sum_{t,t'} \boldsymbol{H}_h^{T-t\top} \tilde{\boldsymbol{W}}_y^\top \boldsymbol{S}_{yx_i} - \boldsymbol{H}_h^{T-t'\top} \tilde{\boldsymbol{W}}_y^\top \tilde{\boldsymbol{W}}_y \boldsymbol{H}_h^{T-t} \tilde{\boldsymbol{W}}_x \boldsymbol{S}_{x_t x_{t'}} \tag{21}$$

$$\frac{\partial \tilde{\boldsymbol{W}}_y}{\partial \tau} = -\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{W}}_y} = \sum_{t,t'} \boldsymbol{S}_{yx_t} \tilde{\boldsymbol{W}}_x^\top \boldsymbol{H}_h^{T-t\top} - \tilde{\boldsymbol{W}}_y \boldsymbol{H}_h^{T-t} \tilde{\boldsymbol{W}}_x \boldsymbol{S}_{x_t x_{t'}} \tilde{\boldsymbol{W}}_x^\top \boldsymbol{H}_h^{T-t'\top}. \tag{22}$$

We note that Eqs. 21-22 hold for both non-normal dynamics and normal dynamics. Moving forward, though, we restrict our attention to the case of normal dynamics (diagonal H_h).We also now make the same diagonalized matrix assumptions in Proca et al. (2025), which is that $\tilde{W}_x$ and $\tilde{W}_y$ have only diagonal entries. This yields update equations where $\boldsymbol{H}$ can combine

$$\frac{\partial \tilde{\boldsymbol{W}}_x}{\partial \tau} = -\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{W}}_x} = \sum_{t,t'} \boldsymbol{H}_h^{T-t} \tilde{\boldsymbol{W}}_y^\top \boldsymbol{S}_{yx_i} - \boldsymbol{H}_h^{2T-t'-t} \tilde{\boldsymbol{W}}_y^\top \tilde{\boldsymbol{W}}_y \tilde{\boldsymbol{W}}_x \boldsymbol{S}_{x_t x_{t'}} \tag{23}$$

$$\frac{\partial \tilde{\boldsymbol{W}}_y}{\partial \tau} = -\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{W}}_y} = \sum_{t,t'} \boldsymbol{S}_{yx_t} \tilde{\boldsymbol{W}}_x^\top \boldsymbol{H}_h^{T-t} - \tilde{\boldsymbol{W}}_y \boldsymbol{H}_h^{2T-t-t'} \tilde{\boldsymbol{W}}_x \boldsymbol{S}_{x_t x_{t'}} \tilde{\boldsymbol{W}}_x^\top. \tag{24}$$

Rather than track how updates for the entire weight matrices unfold under time, it is useful to consider how their columns and vectors, or "modes", of these matrices update over time (Saxe et al., 2014; Proca et al., 2025). Specifically, we define the columns of $\tilde{\boldsymbol{W}}_x$ as $\boldsymbol{a}_\alpha$, and the rows of $\tilde{\boldsymbol{W}}_y$ as $\boldsymbol{c}_\beta$. The diagonal entries of $\boldsymbol{H}_h$ are given by b_α (the eigenvalues of W_h), and similarly $s^\alpha_{x_t y}$ and $s^\alpha_{x_t x_{t'}}$ are the diagonal entries of the task covariance matrices. The modes are then given as

$$\boldsymbol{a}(\tau) = \tilde{\boldsymbol{W}}_{x:,\alpha} = \sum_\alpha a_\alpha(\tau) \boldsymbol{r}_\alpha, \qquad \boldsymbol{b}(\tau) = \tilde{\boldsymbol{W}}_{y_{\alpha,:}} = \sum_\alpha b_\alpha(\tau) \boldsymbol{r}_\alpha, \tag{25}$$

where $\{\boldsymbol{r}_\alpha\} \in R^{N_h \times 1}$ is a basis set of vectors for the modes.

By tracking the α-th columns of $\tilde{W}_x$ and α-th rows of $\tilde{W}_y$ in eqs. 23-24, we can express the update equations for the input and output modes:

$$\begin{aligned}\frac{\partial \boldsymbol{a}_\alpha}{\partial t} &= \sum_{t,t'} \sum_\gamma b_\gamma^{(T-t)} \boldsymbol{c}_\gamma s^\gamma_{yx_i} - b_\alpha^{(T-t)} b_\alpha^{(T-t')} (\boldsymbol{c}_\alpha \cdot \boldsymbol{a}_\alpha) s^\alpha_{x_t x_{t'}} \\ &= \sum_{t,t'} b_\alpha^{(T-t)} \boldsymbol{c}_\alpha \left[s^\alpha_{yx_t} - b_\alpha^{(T-t')} (\boldsymbol{c}_\alpha \cdot \boldsymbol{a}_\alpha) s^\alpha_{x_t x_{t'}} \right] - \sum_{\gamma \neq \alpha} b_\gamma^{(T-t)} \boldsymbol{c}_\gamma s^\gamma_{yx_t}\end{aligned} \tag{26}$$

$$\frac{\partial \boldsymbol{c}_\alpha}{\partial t} = \sum_{t,t'} b_\alpha^{(T-t)} \boldsymbol{a}_\alpha \left[s^\alpha_{yx_t} - b_\alpha^{(T-t')} (\boldsymbol{c}_\alpha \cdot \boldsymbol{a}_\alpha) s^\alpha_{x_t x_{t'}} \right] - \sum_{\gamma \neq \alpha} b^{\gamma^{(T-i)}} \boldsymbol{a}^\gamma s^\gamma_{yx_i} \tag{27}$$

[4] This was also noted in Saxe et al. (2014) Appendix

Eqs. 26-27 contain contributions from their own mode α, as well as cross term from other modes γ. Analytical solutions for this form are not generally tractable because of the contribution from all modes to learning, and so to address this we restrict our analysis to a special set of initial conditions to remove the cross-mode contribution. This is performed by initializing modes in a distinct set of non overlapping basis set vectors $\{\boldsymbol{r}_\alpha\} \in \mathbb{R}^{N_h \times 1}, \boldsymbol{r}_\alpha \cdot \boldsymbol{r}_\beta = \delta_{\alpha\beta}$: As has been shown previously (Saxe et al., 2014), if $\boldsymbol{a}$ and $\boldsymbol{c}$ are initialized onto the same set of orthogonal modes $\{\boldsymbol{r}_\alpha\}$, then we can track the evolution of the coefficients on these modes, and importantly, any interaction terms among these modes are strictly zero with these initial conditions.

The update equations for the weighting coefficients are then given as

$$\frac{\partial a_\alpha}{\partial \tau} = \sum_{t,t'} b_\alpha^{(T-t)} c_\alpha \left[s^\alpha_{yx_t} - b_\alpha^{(T-t')} c_\alpha a_\alpha s^\alpha_{x_t x_{t'}} \right] \tag{28}$$

$$\frac{\partial c_\alpha}{\partial t} = \sum_{t,t'} b_\alpha^{(T-t)} a_\alpha \left[s^\alpha_{yx_t} - b_\alpha^{(T-t')} c_\alpha a_\alpha s^\alpha_{x_t x_{t'}} \right] \tag{29}$$

Because we ignored the mode cross terms, the updates in Eqs. 28-29 minimize an effective loss function, which can be seen by integrating them with respect to the mode coefficients:

$$E = \sum_\alpha \sum_{t,t'}^{T} \left[a_\alpha c_\alpha b_\alpha^{T-t} s^{\alpha^{1/2}}_{x_t x_{t'}} - s^\alpha_{yx_t} s^{\alpha^{-1/2}}_{x_t x'_t} \right] \left[a_\alpha c_\alpha b_\alpha^{T-t'} s^{\alpha^{1/2}}_{x_t x_{t'}} - s^\alpha_{yx_t} s^{\alpha^{-1/2}}_{x_t x_{t'}} \right]. \tag{30}$$

The product $a_\alpha c_\alpha$ has symmetry condition of this energy ($a_\alpha c_\alpha = [a_\alpha / k][c_\alpha k]$), which guarantees an invariance condition $a^2 = c^2$ (Saxe et al., 2014). Moving forward, we omit the α index unless it is strictly necessary. Introducing a collective network parameter $u = ac$, the collective update equation follows a similar functional form

$$\frac{\partial u}{\partial t} = \frac{\partial a}{\partial \tau} c + a \frac{\partial c}{\partial \tau} = 2 \sum_{i,j} b^{(T-t)} u \left[s_{yx_t} - b^{(T-t')} u s_{x_t x_{t'}} \right] \tag{31}$$

where we used the equivalence $a^2 = c^2$. Eq. 31 is a separable differential equation with a closed form solution. To simplify notation, we collect the effect of recurrence and task covariances into terms β_{yx} and β_{xx}

$$\beta^\alpha_{yx} = \sum_t^T b_\alpha^{(T-t)} s^\alpha_{yx_t}, \qquad \beta^\alpha_{xx} = \sum_{t,t'}^T b_\alpha^{(2T-t-t')} s^\alpha_{x_t x_{t'}} \tag{32}$$

The separable equation is then $\frac{\partial u}{\partial t} = 2u \left[\beta_{yx} - u \beta_{xx} \right]$, which in its partial fraction decomposed form is

$$t = \frac{1}{2\beta_{yx}} \int_{u_0}^{u_f} \frac{du}{u} - \frac{1}{2\beta_{yx}} \int_{u_0}^{u_f} \frac{du}{u - \beta_{yx}/\beta_{xx}} \tag{33}$$

Integration of Eq. 33 yields

$$t = \frac{1}{2\beta_{yx}} \log |u|_{u_0}^{u_f} - \frac{1}{2\beta_{yx}} \log |u - \beta_{yx}/\beta_{xx}|_{u_0}^{u_f} \tag{34}$$

which upon reorganization gives the solution for training dynamics $u(\tau) = a(\tau)c(\tau)$

$$a(\tau)c(\tau) = \frac{1}{\left[\frac{1}{(a(0)c(0))} - \frac{\beta_{xx}}{\beta_{yx}} \right] e^{-2\tau \beta_{yx}/\gamma} + \frac{\beta_{xx}}{\beta_{yx}}}, \tag{35}$$

where γ is the inverse of the learning rate. Eq. 35 is for a single mode α, and there will be equivalent expressions for each of the $\alpha = \{1, 2, .. \min[N_x, N_y]\}$ modes. Finally, the optimal solution is found for $t \to \infty$ in Eq. 35,

$$u^{*(k)} = \beta^{(k)}_{yx} / \beta^{(k)}_{xx} = \frac{\sum_i^T b^{T-t} s^{(k)}_{yx_t}}{\sum_{t,t'}^T b^{2T-t-t'} s^{(k)}_{x_t x_{t'}}}. \tag{36}$$

We note that in the special case of constant singular values and perfectly stable dynamics ($b = 1$), this recapitulates the results in Saxe et al. (2014) (Appendix A.6.4).

In summary, we provided an expression for the learning dynamics of input and output modes in linear RNNs that encompass tasks with temporally correlated inputs. The primary assumptions that limit our current approach are 1) requiring input variability to be aligned with input-to-target mappings, 2) assuming normal recurrent dynamics, and most importantly 3) requiring that initial conditions of task parameters are in an orthogonal space with respect to the task geometry to facilitate factorized training with individual network modes.

A.2 Expansion of CL training time improvement

Here we add in a few intermediate steps to show the our core results of the conditions for training time improvement. Starting from the general condition:

$$t_{i\to 2} > t_{i\to 1} + t_{1\to 2}. \tag{37}$$

we first expand the the optimal solutions and initial conditions in terms of the recurrence-weighted singular values for all except solo task 1 and task 2 training,:

$$\begin{aligned}
&\frac{\gamma}{2\beta_{yx}^{(2)}}\left(\log\left|\frac{u^{*(2)}}{u_0}-1\right|-\log\left|\frac{\epsilon^{(2)}}{1-\epsilon^{(2)}}\right|\right) > \\
&\frac{\gamma}{2\beta_{yx}^{(1)}}\left(\log\left[\frac{u^{*(1)}}{u_0}-1\right]-\log\left|\frac{\epsilon^{(1)}}{1-\epsilon^{(1)}}\right|\right) + \\
&\frac{\gamma}{2\beta_{yx}^{(2)}}\left(\log\left|\frac{\beta_{yx}^{(2)}/\beta_{xx}^{(2)}}{(1-\epsilon^{(1)})\beta_{yx}^{(1)}/\beta_{xx}^{(1)}}-1\right|-\log\left|\frac{\epsilon^{(2)}}{1-\epsilon^{(2)}}\right|\right)
\end{aligned} \tag{38}$$

Removing the common term (learning rate, accuracy term for task 2), and bringing all terms to both sides gives

$$\begin{aligned}
&\frac{1}{\beta_{yx}^{(2)}}\left(\log\left|\frac{u^{*(2)}}{u_0}-1\right|\right)-\frac{1}{\beta_{yx}^{(1)}}\left(\log\left[\frac{u^{*(1)}}{u_0}-1\right]-\log\left|\frac{\epsilon^{(1)}}{1-\epsilon^{(1)}}\right|\right) \\
&-\frac{1}{\beta_{yx}^{(2)}}\left(\log\left|\frac{\beta_{yx}^{(2)}/\beta_{xx}^{(2)}}{(1-\epsilon^{(1)})\beta_{yx}^{(1)}/\beta_{xx}^{(1)}}-1\right|\right) > 0
\end{aligned} \tag{39}$$

Finally, multiplying everything by $\beta_{yx}^{(2)}$ and combining the logs of the middle term gives our final expression

$$\log\left|\frac{u^{*(2)}}{u_0}-1\right|+\frac{\beta_{yx}^{(2)}}{\beta_{yx}^{(1)}}\log\left|\left(\frac{\epsilon^{(1)}}{1-\epsilon^{(1)}}\right)\left(\frac{u_0}{u^{*(1)}-u_0}\right)\right|-\log\left|\frac{1}{(1-\epsilon^{(1)})}\frac{\beta_{yx}^{(2)}}{\beta_{yx}^{(1)}}\frac{\beta_{xx}^{(1)}}{\beta_{xx}^{(2)}}-1\right|>0 \tag{40}$$

A.3 Geometry of saddles

Here we detail the fixed-point structure in linear RNNs for factorized tasks. Our approach is based on Baldi & Hornik (1989), which considered equal numbers of inputs and outputs, and we follow suit by considering square matrices for $\boldsymbol{W}_x$, $\boldsymbol{W}_h$, and $\boldsymbol{W}_y$. We also assume in this section that we are in a rotated reference frame such that $\boldsymbol{W}_h = \Lambda$. We will remind the reader of this and our other assumptions the end of the section. We formalize the analysis for factorized tasks, as a key ingredient in this derivation is that the SVD of Σ_{yx_t} has constant eigenvectors in time, an assumption satisfied in factorized tasks that is violated in compositional ones. The gradients of the general (non-factorized) loss function are the following:

$$\frac{\partial \boldsymbol{W}_x}{\partial \tau} = \sum_{t,t'} \boldsymbol{\Lambda}^{T-t}\boldsymbol{W}_y^\top\left(\boldsymbol{\Sigma}_{yx_t} - \boldsymbol{W}_y\boldsymbol{\Lambda}^{T-t'}\boldsymbol{W}_x\boldsymbol{\Sigma}_{x_t x_{t'}}\right) \tag{41}$$

$$\frac{\partial \boldsymbol{W}_y}{\partial \tau} = \sum_{t,t'} \left(\boldsymbol{\Sigma}_{yx_t} - \boldsymbol{W}_y \boldsymbol{\Lambda}^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t x_{t'}} \right) \boldsymbol{W}_x^\top \boldsymbol{\Lambda}^{T-t'}. \tag{42}$$

The existence of saddles is motivated from previous work in auto-encoders and feedforward networks (Baldi & Hornik, 1989), which in our work leads to locating conditions where the network captures only subsets of modes of the input-target covariance in eqs. 41-42. $\boldsymbol{S}_{yx_t}$ has stationary SVD modes for the factorized setting, so we focus on this case, but we start general and note where the analysis becomes complicated for the full case. Turning to a case with N factorized modes, recall that

$$\boldsymbol{\Sigma}_{yx_t} = \boldsymbol{V}_{xy} \boldsymbol{S}_{yx_t} \boldsymbol{U}_{xy}^\top = \sum_n^N s_{yx_t}^{(n)} (\boldsymbol{v}_n \boldsymbol{u}_n^\top) \tag{43}$$

where n denotes each distinct, factorized mode. The global minima will occur when the term in parentheses in eqs. 41-42 is zero, which is when the network successfully captures all of the modes of the input-target covariance. This occurs when

$$\sum_n^N s_{yx_t}^{(n)} (\boldsymbol{v}_n \boldsymbol{u}_n^\top) = \boldsymbol{W}_y \boldsymbol{\Lambda}^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t, x_{t'}} \tag{44}$$

$$\sum_n^N (\boldsymbol{v}_n \boldsymbol{e}_n^\top) s_{yx_t}^{(n)} (\boldsymbol{e}_n \boldsymbol{u}_n^\top) = \boldsymbol{W}_y \boldsymbol{\Lambda}^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t, x_{t'}} \tag{45}$$

In the last step we introduced standard normal vectors $\boldsymbol{e}_n = [0, 0, ...1, 0..]$ to easily isolate input and output weights definitions. Moving forward we assume that the input covariance has no across-time co-variability, and $\boldsymbol{\Sigma}_{x_t, x_{t'}} = \boldsymbol{\Sigma}_{x_t} \delta_{t,t'}$.

A manifold of saddles are present when the network aligns with some (but not all) of the N SVD modes in Eq. 43. If we consider that the network is aligned with a nonzero subset of modes $\{i\}$ in eq. 45, then we can define the form of $\boldsymbol{W}_y$ and $\boldsymbol{W}_x$ for saddles:

$$\sum_{k \in \{i\}} (\boldsymbol{v}_k \boldsymbol{e}_k^\top) s_{yx_t}^{(n)} (\boldsymbol{e}_k \boldsymbol{u}_k^\top) = \boldsymbol{W}_y \boldsymbol{\Lambda}^{T-t} \boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t} \tag{46}$$

$$\boldsymbol{W}_y = \left(\sum_{k \in \{i\}} \boldsymbol{v}_k \boldsymbol{e}_k^\top \right) D = V_{xy}^{\{i\}} D \tag{47}$$

$$\boldsymbol{\Lambda}^{T-t} = \boldsymbol{S}_{yx_t} \tag{48}$$

$$\boldsymbol{W}_x \boldsymbol{\Sigma}_{x_t} = D^{-1} \left(\sum_{k \in \{i\}} \boldsymbol{e}_k \boldsymbol{u}_k^\top \right)$$

$$\boldsymbol{W}_x = D^{-1} U_{xy}^{\{i\}\top} \boldsymbol{\Sigma}_{x_t}^{-1} \tag{49}$$

$V_{xy}^{\{i\}}$ and $U_{xy}^{\{i\}}$ respectively denote rank-deficient versions of V_{xy} and U_{xy} that only contain eigenvectors from the set $\{i\}$, and D is a diagonal matrix that defines the manifold of saddles, since network output is invariant to scaled changes between input and output weights:

$$y_t = \sum_t \boldsymbol{W}_y H^{T-t} \boldsymbol{W}_x x_t = \sum_t \boldsymbol{W}_y D H^{T-t} D^{-1} \boldsymbol{W}_x x_t. \tag{50}$$

For factorized tasks, which require $U_{xx} = U_{xy}$, the input weights will be aligned with the directions of input covariance:

$$\boldsymbol{W}_x = D^{-1} U_{xy}^{\{i\}\top} \boldsymbol{S}_{x_t}^{-1} \tag{51}$$

To see how this produces a saddle, plugging eqs. 47-49 into the gradient expressions in Eqs. 41-42 shows how only orthogonal modes remain from the term in parentheses, leading to zero gradients. we show it here for the gradient of $\boldsymbol{W}_x$ [5]:

$$\begin{aligned}\frac{\partial \boldsymbol{W}_x}{\partial \tau} &= \sum_t \sum_{k\in\{i\}} s^{(k)}_{yx_t} D(\boldsymbol{e}_k \boldsymbol{v}_k^\top) \left(\sum_n^N s^{(n)}_{yx_t} (\boldsymbol{v}_n \boldsymbol{e}_n^\top)(\boldsymbol{e}_n \boldsymbol{u}_n^\top) - s^{(k)}_{yx_t} (\boldsymbol{v}_k \boldsymbol{e}_k^\top) D D^{-1} (\boldsymbol{e}_k \boldsymbol{u}_k^\top) \right) \\ &= \sum_t \sum_{k\in\{i\}} \sum_{k'\notin\{i\}} s^{(k)}_{yx_t} s^{(k')}_{yx_t} D \boldsymbol{e}_k (\boldsymbol{v}_k^\top \boldsymbol{v}_{k'}) \boldsymbol{e}_{k'}^\top \boldsymbol{e}_{k'} \boldsymbol{u}_{k'}^\top \\ &= 0 \end{aligned} \tag{52}$$

The above expression goes to zero due to orthogonality of $\boldsymbol{V}_{xy}$: $\boldsymbol{v}_k^\top \boldsymbol{v}_{k'} = 0$. The same is true for the gradient for $\boldsymbol{W}_y$. The number of saddles is defined by the number of distinct subsets of modes for an $N-$dimensional factorized model (minus the global minima and local maxima), which is $2^N - 2$ saddles.

To derive the explicit form of saddles in our toy task in the main text, and to illustrate how the locations of saddles relative to eigenvalues of the ground-truth recurrence matrix, consider the ground-truth linear RNN with factorized parameters $\{\tilde{\boldsymbol{W}}^*_x = \mathbb{I}, H, \tilde{\boldsymbol{W}}^*_y\}$, where $\tilde{\boldsymbol{W}}^*_y$ is diagonal. The input-target covariance can then be re-expressed as

$$\Sigma_{yx_t} = \sum_t \tilde{\boldsymbol{W}}^*_y \boldsymbol{\Lambda}^{T-t} \tilde{\boldsymbol{W}}^*_x \tag{53}$$

$$\boldsymbol{V}_{xy} = \mathbb{I} \tag{54}$$

$$\boldsymbol{S}_{yx_t} = \tilde{\boldsymbol{W}}^*_y \boldsymbol{\Lambda}^{T-t} \tag{55}$$

$$\boldsymbol{U}_{xy} = \mathbb{I} \tag{56}$$

When directly plugging these into the expressions for the parameters in eqs. 47-49, there is a discrepancy for Λ, which we account for by adjusting the weight of $\boldsymbol{W}_y$. For our 2-D system in the main text, there are 2 saddles, indexed by j,

$$\boldsymbol{W}^{(sj)}_y = \tilde{\boldsymbol{W}}^* \mathbf{I}^{\{j\}} D \tag{57}$$

$$\boldsymbol{W}^{(sj)}_y = D^{-1} \mathbf{I}^{\{j\}} \tag{58}$$

$$\begin{aligned} \boldsymbol{W}^{s1}_y &= \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} D, \quad \boldsymbol{W}^{s1}_x = D^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ \boldsymbol{W}^{s2}_y &= \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} D, \quad \boldsymbol{W}^{s2}_x = D^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{59}$$

Looking at eq. 55, the values of the loss function at each saddle manifold will be dictated by the residual eigenvalue modes that the model fails to capture ($\{\lambda_k\} | k \notin \{i\}\}$), demonstrating that saddles at lower loss values exists for models that capture longer timescale modes, and progressively larger losses as smaller eigenvalues are captured by the model.

In summary, we showed the geometry of saddles for tasks with equal numbers of input and outputs for factorized tasks, and restricted our analysis to inputs without temporal correlations. These results do not hold for compositional tasks because they break that requirements of static SVD eigenvectors. However, in our simulations we found effects of slower learning near the locations of the original saddle manifolds from factorized tasks. We observed that these former saddles were replaced by a manifold of slow "ghost" points that still reflected some of the saddle-point structure, but were not as distinct as in the factorized setting.

[5] Multiplying $\boldsymbol{W}_y$, $\boldsymbol{\Lambda}^{T-t}$, and $\boldsymbol{W}_x$ will only contain a single index k due to the orthogonality of the standard normal basis $\{\boldsymbol{e}\}$.

A.4 SIMULATION AND TRAINING METHODS

All simulations were performed in Python. Numerical simulations of factorized tasks in Section 3 were performed using gradient descent using a learning rate of 0.001 unless otherwise noted, and used custom linear RNN networks with with 128 units and a single input and output channel. All initial conditions were initialized into a single input-target mode as specified by Eq. 25 with the value $a = c = 0.01$. In practice, this meant setting a single entry $W_x[0, 1]$ and $W_y[0, 1]$ to 0.01, with the rest of the values initialized to zero. The recurrent weights were set to a diagonal matrix where all eigenvalues were $b = 0.96$. Compositional studies in Section 4 used the same RNN structure with 2 input/output channels, but with 2 hidden units with eigenvalues $\lambda = 0.6, 0.7$, and a learning rate of 0.01.

For linear RNN simulations fitting that compared learning trajectories to theory, we first calculated the theoretical optimal solution using Eq. 36, then trained the network until it converged to weights in an ϵ window of the optimal value. To calculate this optimal value, and to produce the theoretical wait time predictions and learning trajectory curves, we used analytically defined covariance matrices based on the defined task structure.

For nonlinear RNNs in Appendix A.6.3, we approximated the optimal solution by training the RNN until convergence, and then calculated the numerical training time in each task by retraining the network from scratch until it reached an ϵ window of the optimal solution.

When generating phase portraits in Fig. 2 and results for nonlinear RNNs across a range of task $\mathcal{T}_1$ accuracies in Appendix A.6.3, we only modulated one task parameter at a time, and kept all parameters across $\mathcal{T}_1$ and $\mathcal{T}_2$ equal. When modulating covariance strength $\boldsymbol{S}_{x_t y}$, we held $\boldsymbol{S}_{x_t x_{t'}} = 1.0$. Similarly, when modulating $\boldsymbol{S}_{x_t x_{t'}}$, we held $\boldsymbol{S}_{x_t y}$=1.0 In sec. A.6.1, when modulating K we held κ=0.5; when modulating κ, we held $K = 0.5$.

A.5 CL TASK DEFINITIONS

We studied many different forms of curricula when examining compositional tasks. In all cases the target input channel was the identity, as previously mentioned. Below are the explicit $\boldsymbol{W}_y$ targets for the pretraining CL task. The scale factor C is the constant by which we scaled this target when studying amplitude effects:

$$\boldsymbol{W}_y = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \quad \text{(factorized task)} \tag{60}$$

$$\boldsymbol{W}_y = \begin{bmatrix} C & 0 \\ C & C \end{bmatrix} \quad \text{(scaled target task)} \tag{61}$$

$$\boldsymbol{W}_y = \begin{bmatrix} 0 & C \\ C & C \end{bmatrix} \quad \text{(Task permutation 1. light blue)} \tag{62}$$

$$\boldsymbol{W}_y = \begin{bmatrix} C & C \\ 0 & C \end{bmatrix} \quad \text{(Task permutation 2. medium blue)} \tag{63}$$

$$\boldsymbol{W}_y = \begin{bmatrix} C & C \\ C & 0 \end{bmatrix} \quad \text{(Task permutation 3. dark blue)} \tag{64}$$

$$\boldsymbol{W}_y = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} \quad \text{(low rank task 1. light pink)} \tag{65}$$

$$\boldsymbol{W}_y = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \quad \text{(low rank task 2. dark pink)} \tag{66}$$

$$\boldsymbol{W}_y = \begin{bmatrix} \mathcal{N}(C, 1) & \mathcal{N}(C, 1) \\ \mathcal{N}(C, 1) & \mathcal{N}(C, 1) \end{bmatrix} \quad \text{(random task)} \tag{67}$$

A.6 SUPPLEMENTAL ANALYSES

A.6.1 TEMPORAL CORRELATIONS IN TASK COVARIANCES

To see how the temporal properties of the task can support this, we study correlated inputs generated by an AR1 process as

$$x_t = Kx_{t-1} + w_t, \tag{68}$$

where $w_t \sim \mathcal{N}(0, \Sigma_0)$ is white noise with covariance Σ_0, and K defines the strength of the temporal correlation. For this process, input covariances depend only upon the lag between time-points (Fig. 2A) [6]. Here we consider integration tasks that do not simply perform perfect integration across all time, but are instead leaky integration tasks that weight later time points in a trial

$$y_t = \sum_t^T A_t x_t, \quad A_t = A_0 e^{-\kappa(t-T)} \tag{69}$$

where A_0 is the $N_y \times N_x$ matrix that mixes inputs to output channels, and κ is the decay of the integration profile

[6] correlations that are purely lag-dependent hold only in the infinite-time limit, and we account for finite time correlations when we calculate $\Sigma_{x_t x_t'}$.

A.6.2 MISMATCHED ACROSS-TASK GEOMETRY

In order to study an extension beyond our theory for cases in which the factorized modes of task 1 no longer align with task covariances in task 2, we simulated a system with two inputs and two outputs, which gives two factorized modes in the network. We focused on examining if relative changes in input-target covariance strength would be recapitulated even in this scenario, by seeing if training times reflected the results Fig. 2A-C. The target task $\mathcal{T}_2$ here had $\boldsymbol{S}_{x_t y} = \text{diag}[1.2, 1.0]$ for its input-target covariance singular values, and $\boldsymbol{S}_{x_t x_{t'}} = \text{diag}[1.2, 1.0]$ for its input-input covariances singular values. To avoid any degeneracies in the network we set the two related recurrence eigenvalues to $\lambda = [0.96, 0.94]$, and set the remaining values to 0. The desired accuracy for both tasks was set to 95%. $\boldsymbol{S}_{x_t y}$ for task 1 always kept one mode fixed to the same value for $\mathcal{T}_2$, but we set its second singular value to be either larger or smaller. The left eigenvectors for $\boldsymbol{\Sigma}_{x_t y}$, U_{xy} (Eq. 18), were set to the identity. Importantly, we looked at training times in the "geometry-matched" regime where U_{xy} was also the identity, as well as a "geometry-mismatched" regime in which U_{xy} was rotated with a 2D rotation matrix by $\pi/4$. We found that the qualitative trend seen in Fig. 2A-C holds, which is that relatively stronger $\mathcal{T}_1$ $\boldsymbol{S}_{x_t t}$ strength led to faster training, and vice versa. Moreover, we found that there was some representational rotational effects that occurred for $\mathcal{T}_1$ having weaker input-target covariance strength, which we measured by calculating the normalized inner product between the input matrix W_x over training to its final solution at the end of $\mathcal{T}_2$.

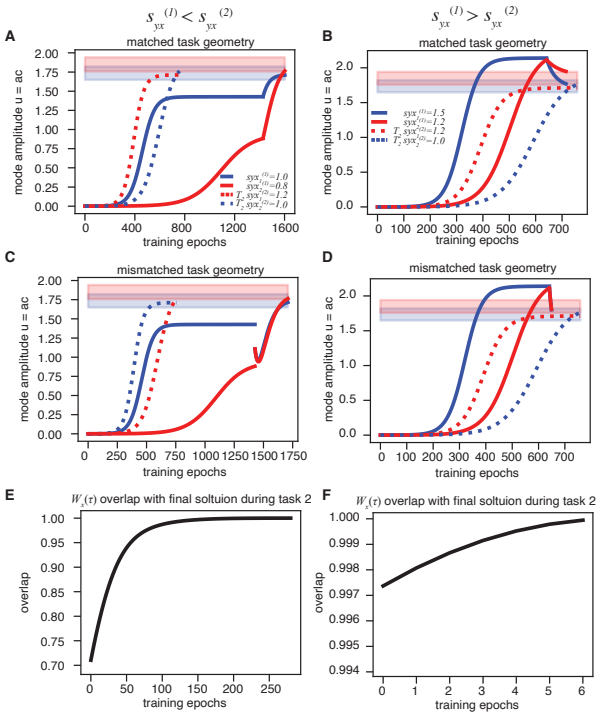

Figure S.1: Effects of across-task, mismatched geometry. Training time in a 2D system with varying input-target covariances was studied when covariance eigenvalues for $\mathcal{T}_1$ and $\mathcal{T}_2$ were aligned, or mismatched. Left column is for the case where one input-target singular value is smaller in $\mathcal{T}_1$ than in $\mathcal{T}2$, and vice versa in the right column. **A-B** Training for the CL sequence vs. direct training when task geometry is matched. Blue lines correspond to the mode that has singular value strength that is matched across the tasks, and red lines denote the one that is different. Dotted lines denote direct training of $\mathcal{T}_2$, solid lines denote the CL sequence training. **C-D** Training for the CL sequence vs. direct training when task geometry is mis-matched. **D-E** Overlap of the input weights throughout training $W_x(\tau)$ with the final solution for input weights $W_x(\tau_f)$. Shaded regions in panels **A-D** denote the one-sided 95% accuracy level around the theoretical solution for each factorized mode.

A.6.3 NONLINEAR RNNS

We also wished to see if the insights found in our linear RNN analysis would hold in a more practical scenario. So we performed the same CL studies, using the same integration tasks, but with RNNs containing a ReLU nonlinearity. For individual RNNs, we compared the training time for networks that had either a linear or ReLU activation function, and we investigated four different parameter regimes that characterize the main aspects of task covariances (Fig. S.2). Without the ability to generate theoretical predictions for nonlinear networks, we instead focused on whether or not the same qualitative principles identified in our linear theory would hold in nonlinear networks. While there are numerical differences in the optimization time, we found evidence that the same qualitative trends seen for linear RNNs can hold even for nonlinear networks, meaning the the relative amplitude of task covariance strength and temporal correlations between two tasks appears to hold.

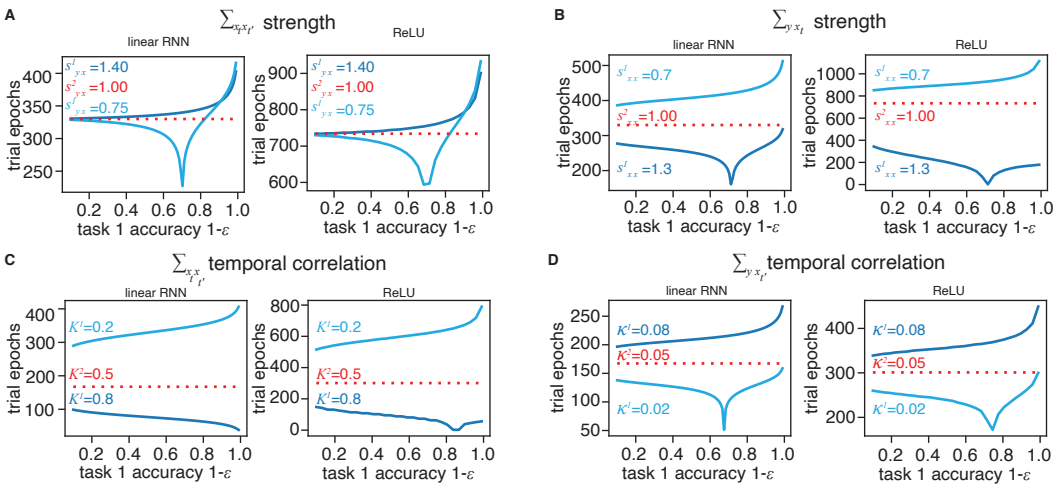

Figure S.2: CL effects in nonlinear RNNs compared to linear RNN theoretical predictions. Total training times for individual RNNs across a range of $\mathcal{T}_1$ accuracies for two different task covariance parameter settings: Red lines denote directly training on $\mathcal{T}_2$ to 99% accuracy and blue lines denote a CL sequence training on $\mathcal{T}_1$ first, with either a larger (dark blue) or smaller (light blue) parameter. Parameter values for each scenario are provided as legends. **A-B** Modulating input-target covariance strength as in Fig. 2**A**-**D**. **C-D** Modulating temporal correlations in task covariance as in Figure 2**E**-**H**

A.6.4 CL EFFECTS WITHOUT RECURRENCE

To make a comparison to feed-forward networks, we also simulated the theoretical results of varying covariance strength for the input-target covariance when the recurrent weights were set equal to the identity matrix. As mentioned in the main text, in this special case of constant singular values and perfectly stable dynamics ($b = 1$), this recapitulates the theoretical results of feed-forward networks studied in Saxe et al. (2014). Here we demonstrate that the same trends in CL improvement exist in this setting, though the magnitude of the effect is reduced.

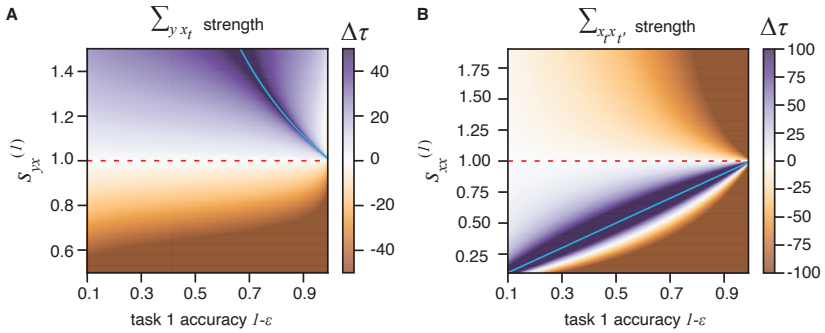

Figure S.3: Theoretical predictions for varying input-target covariance strength with perfect recurrence (b=1). Results are as in Fig. 2A,2D.

A.6.5 JOINT TRAINING WITH RECURRENCE

We also sought to understand if our qualitative results about CL effectiveness could hold when jointly training recurrent weights alongside input and output weights. Introducing training of recurrent weights is more numerically unstable then just input and output weight training because of exploding/vanishing gradient issues; in order to make a tractable comparison we focused on initial conditions near an optimum that we had originally found when training the inputs and outputs, then saw how the parameters changed from there. Specifically, we looked at scenarios analogous to Figure 2A-C, where we modulated the strength of input-target covariance strength. We trained systems with 100 timesteps with a much smaller learning rate ($\gamma = 10^{-7}$), and we chose initial conditions where the single nonzero eigenvalue of the recurrent weight matrix was set to $b = 0.96$, which was the value used in the rest of our work. We then scaled down the optimal input and output values by 50%, then trained all parameters to a loss $\mathcal{L} < 10^{-5}$.

We found that the inputs and outputs barely changed in magnitude, and that the recurrent weight changes dominated the optimization in scenarios where task 1 had a weaker input-target covariance (Fig. S.4A), or a stronger one (Fig. S.4B). We did find that the qualitative trend where stronger task 1 covariance strength coincided with faster training due to curriculum learning, and longer training coincided with a weaker covariance in task 1.

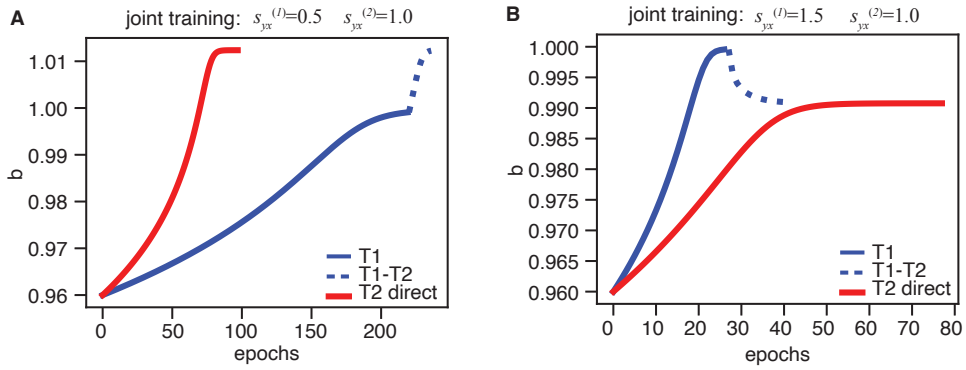

Figure S.4: Recurrent weight change over joint training of all network parameters. Input and output weights were nearly constant for the entire optimization, and are not shown. **A** Training where $\mathcal{T}_1$ had a weaker input-target covariance strength than $\mathcal{T}_2$.**B** Training where $\mathcal{T}_1$ had a stronger input-target covariance strength than $\mathcal{T}_2$. Red lines denote direct task 2 training, and blue lines denote CL sequence training. Blue dotted line shows where $\mathcal{T}_2$ training begins.

A.6.6 NUMERICAL COMPARISONS TO THEORY IN FACTORIZED TASKS

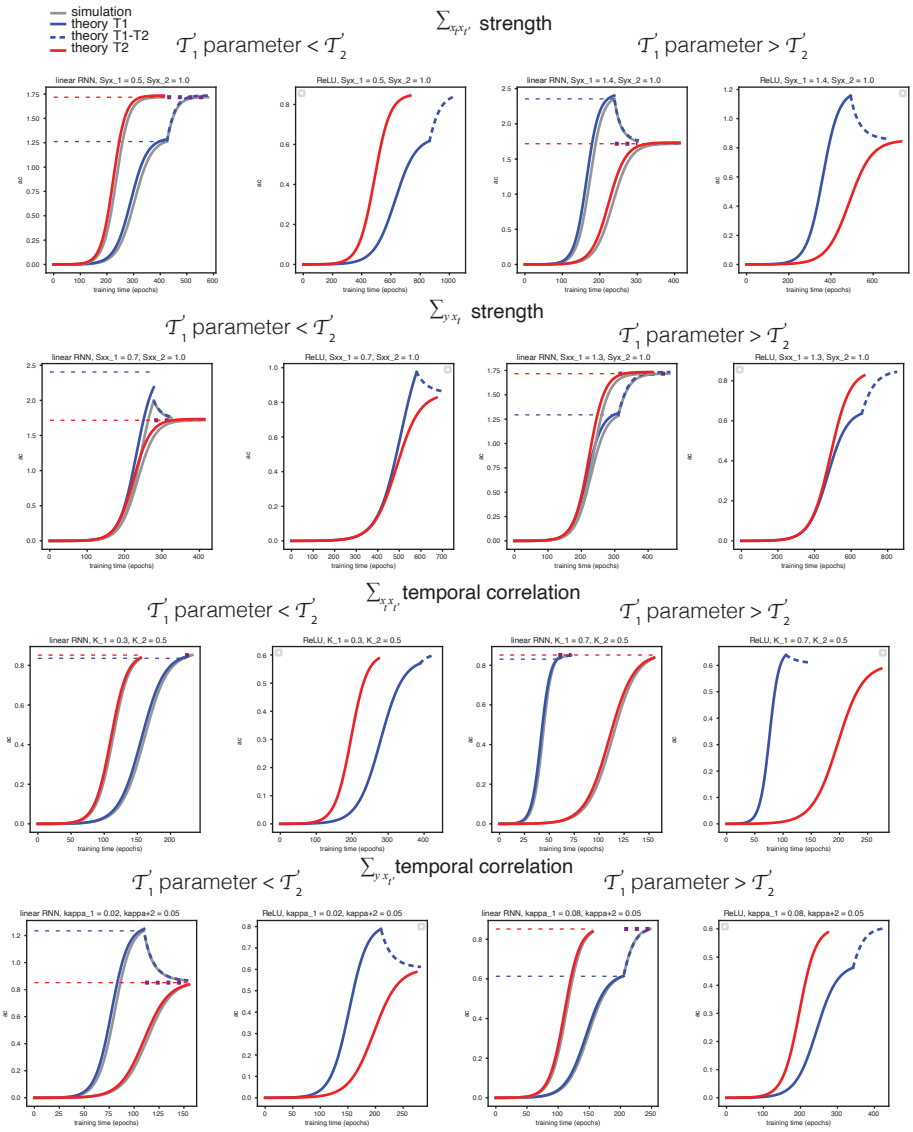

Figure S.5: Numerical fits of optimizations using Curriculum learning (blue) vs. direct training of the target task (red). ReLU learning trajectories are also shown. Each row corresponds to manipulating one aspect of task covariance structure, and here we provide examples for both a parameter setting in task 1 that is smaller than task 2 (left 2 plots) as well as larger (right two plots). Theoretical optima are shown for each task with horizontal dotted lines

A.6.7 FULL COMPOSITIONAL CL RESULTS

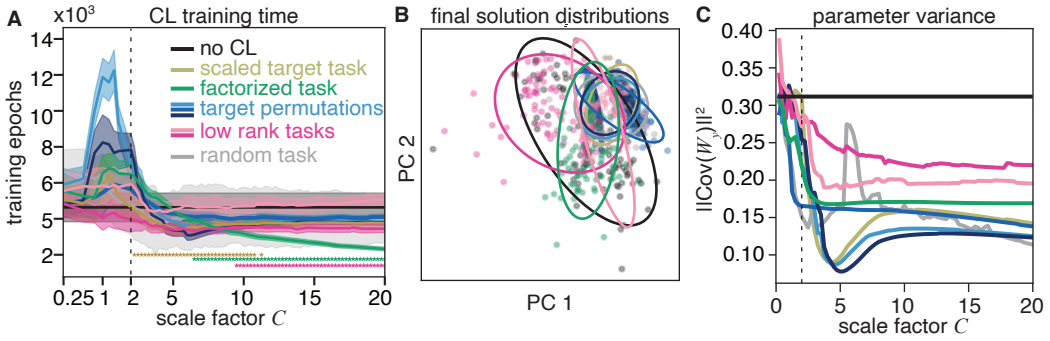

Figure S.6: Results from Fig. 3, but with all CL studied. $* = p < 0.001$ (sign-rank test). Note that one of the low-rank results was significantly different than direct training, though its effect size was much weaker than factorized CL.

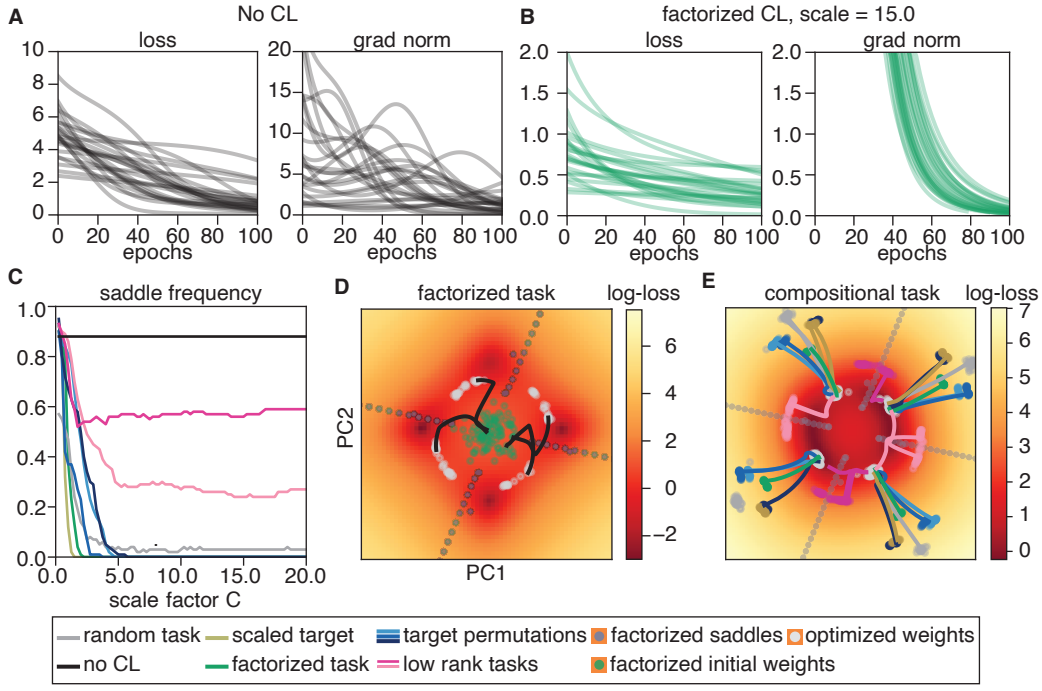

Figure S.7: Results from Fig. 4, but with all CL studied

A.6.8 NORM OF GRADIENTS FOR LOSS LANDSCAPE

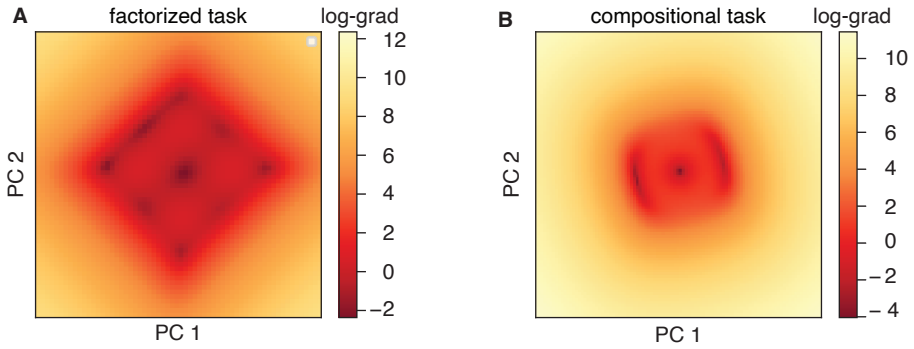

Figure S.8: The norm of the gradient for the PCA phase portrait of the loss in Fig. 4**D**-**E**.