Optimal Protocols for Continual Learning via Statistical Physics and Control Theory

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Artificial neural networks often struggle with *catastrophic forgetting* when learning tasks sequentially, as training on new tasks degrades the performance on earlier ones. Recent theoretical work tackled this issue by analysing learning curves in synthetic settings with predefined training protocols. However, these protocols were heuristic-based and lacked a solid theoretical foundation for assessing their optimality. We address this gap by combining exact training dynamics equations, derived using statistical physics, with optimal control methods. We apply this approach to teacher-student models of continual learning, obtaining a theory for task-selection protocols that optimise performance minimising forgetting. Our analysis offers non-trivial yet interpretable strategies, showing how optimal learning protocols modulate established effects, such as the influence of task similarity on forgetting. We validate our theoretical findings on real-world data.

1 Introduction

Mastering a range of problems is crucial for both artificial and biological systems. In the context of training a neural network on a series of tasks—a.k.a. *multi-task learning* [1, 2, 3, 4]—the ability to learn new tasks can improve leveraging knowledge from previous ones [5]. However, this process can lead to *catastrophic forgetting*, where learning new tasks degrades performance on older ones. This phenomenon has been observed in theoretical neuroscience [6, 7] and machine learning [8, 9], and occurs when the network parameters encoding older tasks are overwritten while training on a new task. Several mitigation strategies have been proposed [10, 11], including semi-distributed representations [12, 13], regularisation methods [14, 15, 16], dynamical architectures [17, 18], and others (see e.g. [19, 20] for thorough reviews). A common strategy, known as *replay*, is to present the network with examples from the old tasks while training on the new one to minimise forgetting [21, 22, 23]. Related theoretical works are discussed in Appendix A. Despite the significant interest in transfer learning and catastrophic forgetting, mitigation strategies considered thus far were pre-defined heuristics, with no guarantees of optimality. In contrast, we aim at identifying the optimal protocol to minimise forgetting. Specifically, we focus on *replay* as a prototypical mitigation strategy and employ control theory to find the optimal training protocol maximising performance across different tasks.

Our contribution. In this work, we combine techniques from statistical physics [24, 25, 26] and Pontryagin's maximum principle from control theory [27, 28, 29] to derive optimal task-selection protocols for the training dynamics of a neural network in a continual learning setting. Pontryagin's principle works efficiently in low-dimensional deterministic systems, hence requiring the statistical physics approach to neural networks [30], where the evolution of high-dimensional stochastic systems are condensed to a few key order parameters governed by ordinary differential equations (ODEs) [24, 25, 26]. Specifically, we consider the teacher-student framework of [31]—a prototype continual learning setting amenable to analytic characterisation. Our main contributions are:

Submitted to the Mathematics of Modern Machine Learning Workshop at NeurIPS 2024. Do not distribute.

- We leverage the ODEs for the learning curves of online SGD to derive closed-form formulae for the optimal training protocols. In particular, we provide equations for the optimal task-selection protocol and the optimal learning rate schedule, as a function of the task similarity γ and the problem parameters. Our framework is broadly applicable beyond the specific context of continual learning, and we outline several potential extensions.
- We evaluate our equations for a range of problem parameters and find highly structured protocols. Interestingly, we are nonetheless able to interpret these strategies a posteriori, formulating a criterion for "pseudo-optimal" task-selection: an initial *focus* phase, where only the new task is presented, followed by a *revision* phase, where old tasks are replayed.
- We clarify the impact of task similarity on catastrophic forgetting. At variance with what observed in [32, 31, 33], forgetting is minimal at intermediate task similarity with optimal task selection. We give a mechanistic explanation of this phenomenon disentangling dynamical effects on the first-layer and readout weights.
- We show that insights from the optimal strategy transfer to real datasets. Specifically, we consider a continual learning task on the Fashion-MNIST dataset, and show that the optimal strategy interpolates between simple heuristic strategies based on the problem's parameters.

2 Model-based theoretical framework

We model supervised learning of multiple tasks. Following [31, 33], we consider a teacher-student framework [34] where a "student" neural network is trained on synthetic inputs $\boldsymbol{x} \in \mathbb{R}^N$, drawn i.i.d. from a Gaussian distribution, $x_i \sim \mathcal{N}(0,1)$. The labels for each task $t = 1, \ldots, T$ are generated by single-layer "teacher" networks: $y^{(t)} = g_*(\boldsymbol{x} \cdot \boldsymbol{w}_*^{(t)}/\sqrt{N})$, where $\boldsymbol{W}_* = (\boldsymbol{w}_*^{(1)}, \ldots, \boldsymbol{w}_*^{(T)})^\top \in \mathbb{R}^{T \times N}$ denote the corresponding teacher vectors, and g_* the activation function. The student is a two-layer neural network with K hidden units, first-layer weights $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K)^\top \in \mathbb{R}^{K \times N}$, activation function g, and second-layer weights $\boldsymbol{v} \in \mathbb{R}^K$, that outputs the prediction:

$$\hat{y}\left(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{v}\right) = \sum_{k=1}^{K} v_k \, g\left(\frac{\boldsymbol{x} \cdot \boldsymbol{w}_k}{\sqrt{N}}\right) . \tag{1}$$

Following a standard *multi-headed* approach to continual learning [15, 35], we allow for task-dependent readout weights: $\boldsymbol{V} = (\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(T)})^\top \in \mathbb{R}^{T \times K}$. While the readout is switched during training according to the task under consideration, the first-layer weights are shared across tasks. A pictorial representation of this model is displayed in Fig. 6 of Appendix D. Training is performed via Stochastic Gradient Descent (SGD) on the squared loss of $y^{(t)}$ and $\hat{y}^{(t)} = \hat{y}(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{v}^{(t)})$, in the *online* regime, where at each training step the algorithmic update is computed using a new sample $(\boldsymbol{x}, y^{(t)})$. The generalisation error of the student on task t is given by

$$\varepsilon_t\left(\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{W}_*\right) := \frac{1}{2}\left\langle \left(y^{(t)} - \hat{y}^{(t)}\right)^2 \right\rangle = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}\left[\left(g^*\left(\frac{\boldsymbol{w}_*^{(t)} \cdot \boldsymbol{x}}{\sqrt{N}}\right) - \hat{y}(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{v}^{(t)})\right)^2\right] , \tag{2}$$

where we use the angular brackets $\langle \cdot \rangle$ to denote the expectation over the input distribution for a given set of teacher and student weights. As shown in [31, 33], we can derive a set of dynamical equations for the generalisation error across training in the high-dimensional limit. We leverage this low-dimensional description and optimal control theory to derive *optimal training protocols* for multi-task learning. In particular, we optimise over task selection and learning rate.

***Forward* training dynamics.** As further discussed in Appendix B, in the limit of large input dimension $N \to \infty$ with $K, T \sim \mathcal{O}_N(1)$, the dynamics of the generalisation error is entirely captured by the evolution of the readouts $\boldsymbol{V}$ and the low-dimensional matrices—a.k.a *overlaps*:

$$M_{kt} := \frac{\boldsymbol{w}_k \cdot \boldsymbol{w}_*^{(t)}}{N} , \qquad Q_{kh} := \frac{\boldsymbol{w}_k \cdot \boldsymbol{w}_h}{N} , \qquad S_{tt'} := \frac{\boldsymbol{w}_*^{(t)} \cdot \boldsymbol{w}_*^{(t')}}{N} , \tag{3}$$

for all $k, h = 1, \ldots, K$ and $t = 1, \ldots, T$. For the remainder of the paper we consider $K = T$, to guarantee that the student network has enough capacity to learn all tasks perfectly. Teacher vectors are normalised, while the task similarity is tuned by a parameter γ, so that $S_{tt'} = \delta_{t,t'} +$

$\gamma(1-\delta_{t,t'})$. For simplicity, it is useful to encode all the dynamical degrees of freedom of interest—the overlaps and the readout weights—in the same vector. We use the shorthand notation $\mathbb{Q} = (\mathrm{vec}(\boldsymbol{Q}), \mathrm{vec}(\boldsymbol{M}), \mathrm{vec}(\boldsymbol{V}))^\top \in \mathbb{R}^{K^2+2KT}$. Following [31], we write a set of ODEs

$$\frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} = f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), \boldsymbol{u}(\alpha)\right) \qquad \text{with } \alpha \in (0, \alpha_F] \,, \tag{4}$$

α denoting the effective training *time*—i.e., the ratio between training epochs and input dimension N, as detailed in Appendix B—and $\boldsymbol{u}$ the dynamical variables that we want to control optimally. In particular, we study the optimal schedules for task-selection $t_c(\alpha)$ and learning rate $\eta(\alpha)$. Here, $t_c(\alpha) \in \{1, \dots, T\}$ indicates on which task the student is trained at time α. The specific form of the functions $f_{\mathbb{Q}}$ is derived in Appendix B. We stress that Eq. 4 is a low-dimensional deterministic equation that fully captures the high-dimensional stochastic dynamics of SGD as $N \to \infty$. This dimensionality reduction is crucial to apply the optimal control techniques in the next section.

Optimal control framework and *backward* conjugate dynamics. Our first main contribution is to derive training strategies that are optimal with respect to the generalisation performance *at the end of training* and on *all tasks*. In practice, the goal of the optimisation process is to minimise a linear combination of the generalisation errors on the different tasks at the final training time α_F:

$$h(\mathbb{Q}(\alpha_F)) = \sum_{t=1}^{T} c_t\, \varepsilon_t(\mathbb{Q}(\alpha_F)) \qquad \text{with} \quad c_t \geq 0 \text{ and } \sum_{t=1}^{T} c_t = 1 \,, \tag{5}$$

where the coefficients c_t identify the relative importance of different tasks and ε_t denotes the infinite-dimensional limit of the average generalisation error on task t, as defined in Eq. 2. Crucially, we have an analytic expression for ε_t, derived in Appendix B. In the remainder of the paper, we assume equally important tasks $c_t = 1/T$. As customary in optimal control theory [28], we adopt a variational approach to solve the problem. We define the cost function

$$\mathcal{F}[\mathbb{Q}, \hat{\mathbb{Q}}, \boldsymbol{u}] = h\left(\mathbb{Q}(\alpha_F)\right) + \int_0^{\alpha_F} \mathrm{d}\alpha\, \hat{\mathbb{Q}}(\alpha)^\top \left[-\frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} + f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), \boldsymbol{u}(\alpha)\right)\right] \,, \tag{6}$$

where the *conjugate order parameters* $\hat{\mathbb{Q}} = (\mathrm{vec}(\hat{\boldsymbol{Q}}), \mathrm{vec}(\hat{\boldsymbol{M}}), \mathrm{vec}(\hat{\boldsymbol{V}}))^\top$ enforce the training dynamics in the training interval $\alpha \in [0, \alpha_F]$. Finding the optimal protocol amounts to minimising the cost function $\mathcal{F}$ with respect to $\mathbb{Q}$, $\hat{\mathbb{Q}}$, and $\boldsymbol{u}$. We defer the details to Appendix B. The minimisation with respect to $\mathbb{Q}$ provides a set of equations for the *backward* dynamics of the conjugate parameters

$$-\frac{\mathrm{d}\hat{\mathbb{Q}}(\alpha)^\top}{\mathrm{d}\alpha} = \hat{\mathbb{Q}}(\alpha)^\top \nabla_{\mathbb{Q}} f_{\mathbb{Q}}(\mathbb{Q}(\alpha), \boldsymbol{u}(\alpha)) \,, \quad \hat{\mathbb{Q}}(\alpha_F) = \nabla_{\mathbb{Q}} h(\mathbb{Q}_F) = \sum_{t=1}^{T} c_t \nabla_{\mathbb{Q}} \varepsilon_t(\mathbb{Q}(\alpha_F)) \,. \tag{7}$$

The optimal control curve $\boldsymbol{u}^*(\alpha)$ is obtained as the solution of the minimisation:

$$\boldsymbol{u}^*(\alpha) = \underset{\boldsymbol{u} \in \mathcal{U}}{\mathrm{argmin}} \left\{ \hat{\mathbb{Q}}(\alpha)^\top f_{\mathbb{Q}}(\mathbb{Q}(\alpha), \boldsymbol{u}) \right\} \,, \tag{8}$$

where $\mathcal{U}$ is the set of allowed controls. For instance, for task selection we take $u(\alpha) = t_c(\alpha)$ and $\mathcal{U} = \{1, 2 \dots, T\}$, where we use $t_c(\alpha)$ to indicate the task on which the student is trained at time α. When optimising over both task selection and learning rate schedule we take $\boldsymbol{u} = (t_c, \eta)$ and $\mathcal{U} = \{1, 2 \dots, T\} \times \mathbb{R}^+$. Crucially, the optimal control equations 4, 7, and 8 must be iterated until convergence, starting from an initial guess on $\boldsymbol{u}$. Let us stress that the space $\mathcal{U}$ of possible controls is high-dimensional and hence it is not feasible to explore it via greedy search strategies.

3 Results and applications

3.1 Experiments on synthetic data

We formulate the continual learning problem as follows. During a first training phase, the student learns perfectly task $t = 1$. Then, the goal is to learn a new task $t = 2$ without forgetting the old one during a second training phase of duration α_F. We investigate the role of *replay*—i.e., using samples from task 1 during the second training phase—and the structure of the optimal replay strategy.

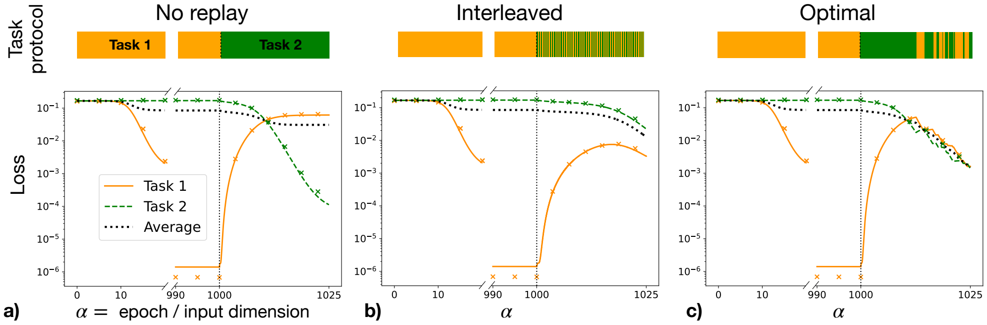

Figure 1: The student is trained on task 1 during the first phase ($\alpha \in [0, 1000]$), then task 2 is introduced. During the second phase ($\alpha \in (1000, 1025]$), task 1 may be replayed to prevent forgetting. For better visibility, we only display the regions $\alpha \in [0, 20] \cup [990, 1025]$. We compare three strategies: **a)** no replay, **b)** interleaved replay **c)** the optimal strategy derived in Sec. 2. Crosses mark numerical simulations of a single trajectory at $N = 20000$, lines mark the solution of Eq. 4. Colour bars represent the protocol t_c. Parameters: $\gamma = 0.3$, $K = T = 2$, and $\eta = 1$.

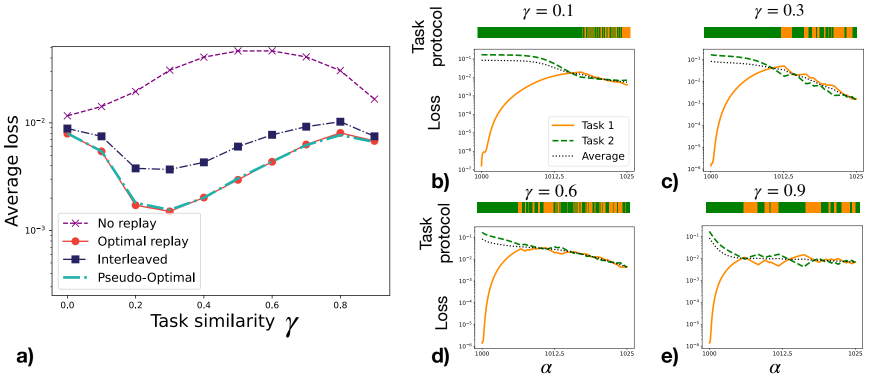

Figure 2: **a)** Average loss at the end of the second training phase as a function of task similarity γ. **b-e)** Optimal replay strategies for different values of γ. Colour bars show the protocol $t_c(\alpha)$.

To this end, we take the task-selection variable as our control $u(\alpha) = t_c(\alpha) \in \{1, 2\}$, while we set $t_c = 1$ during the first training phase. The result of the optimisation in Eq. 8 balances training on the new task with replaying the old task. We do not enforce any constraints on the number of samples from task 1 to use in the second phase, therefore our method provides both the optimal *fraction* of replayed samples and the optimal task *ordering*, depending on the time window α_F. Fig. 1 compares the learning dynamics of three different strategies, depicting the loss on task 1 (orange), task 2 (dashed green), and their average (dotted black) as a function of the training time α. The student is trained exclusively on task 1 until $\alpha = 1000$, when the task is perfectly learned. Then, the student is trained on both tasks for a training time of duration $\alpha_F = 25$. A colour bar above each plot illustrates the associated task-selection strategy $t_c(\alpha)$. Panel **a)** shows that training without replay leads to catastrophic forgetting of task 1. Panel **b)** shows a heuristic "interleaved" strategy, where training alternates one sample from the new task to one from the old one. As observed in [33], the interleaved strategy already improves performance, demonstrating the relevance of replay to mitigate forgetting. Panel **c)** of Fig. 1 shows the loss dynamics for the optimal replay strategy. Notably, this strategy has a complex structure and displays a clear performance improvement over the other two strategies.

The impact of task similarity. We examine the performance of the optimal strategy in relation to task similarity γ. Fig. 2**a)** depicts the average loss at the end of training as a function of γ. For the no-replay strategy, as noted in [31, 33], intermediate task similarity induces the highest error. [33] explained this non-monotonicity as a trade-off between node re-use and node activation. Indeed, for small γ, there is minimal interference between tasks, and one hidden neuron predominantly aligns with the new task, while the other retains knowledge of the old task, leading to *specialisation*. At large γ, features from task 1 are reused for task 2, avoiding forgetting. However, at intermediate γ, interference is maximal, both neurons quickly align with task 2, and task 1 is forgotten. Fig. 2**a)** shows that replay reverses this trend: the minimal error occurs at intermediate γ. To explain this nontrivial behaviour, we must first understand the optimal replay protocol.

Interpretation of the optimal replay structure. The optimal replay dynamics is illustrated in panels **b-e)** of Fig. 2 for different values of γ, displaying a highly structured protocol. We can interpret this structure a posteriori: an initial *focus phase* without replay is followed by a *revision phase* involving interleaved replay. The transition between these two phases corresponds approximately to the point at which the loss on the new task matches the loss on the old one. To investigate the significance of this structure, we also test an interleaved strategy, plotted in Fig. 2**a)**, where task ordering in the second training phase is fully randomised but maintains the same overall replay fraction of the optimal strategy. This protocol has a performance gap compared to the optimal one, showing the importance of a properly structured replay scheme. Additionally, we test a "pseudo-optimal" variant, where the *focus phase* is retained, but the *revision phase* is randomised. This variant performs comparably to the optimal strategy, suggesting that while the specific order of the revision phase is largely unimportant, it is key to precede it with a training phase on the new task.

We now examine the inverted non-monotonic behaviour of the average loss as a function of γ under the optimal protocol. First, as shown in Fig. 7 of Appendix D, the optimal protocol achieves a good level of node specialisation across all values of γ. Thus, replay prevents task interference that typically causes performance deterioration at intermediate γ. The non-monotonic behaviour of the optimal curve in Fig. 2**a)** arises from a different origin, involving two opposing effects related to the first-layer weights and the readout. The initial decrease of the loss with γ is quite intuitive, as only minimal knowledge can be transferred from task 1 to task 2 when γ is small. Consequently, the focus phase on task 2 must be longer for smaller γ, leaving less time to revise task 1, thereby reducing performance. The performance decrease observed in Fig. 2a) for $\gamma > 0.3$ is more subtle and is related to the readout layer. A detailed explanation of this result is provided in Appendix C.

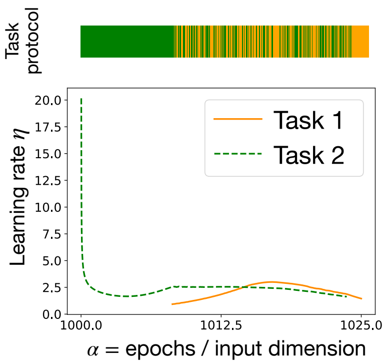

Figure 3: Jointly-optimal task selection and learning rate for the same parameters as Fig. 1. The colour bar marks $t_c(\alpha)$.

Optimal learning rate. Optimal learning rate dynamics have been studied with a similar approach in [36]. Here, we consider the joint optimisation of replay protocol and learning rate. Fig. 3 shows the optimal learning rate schedule for task similarity $\gamma = 0.3$ in the second training phase ($\alpha_F = 25$). Optimal task-selection is again characterised by an initial focus phase, that also coincides with a strong annealing of the learning rate to achieve optimal performance. Interestingly, in the revision phase, the optimal learning rate schedule exhibits a highly nontrivial structure (see Fig. 3). Indeed, although the optimal learning rate curve is unique, we find that effectively it can be seen as two different curves, associated to the respective tasks. In practice, the optimal learning rate curve "jumps" between these two curves according to the task selected at a given training time. Overall, the joint optimisation over task selection and learning rate provides a significant improvement in performance, as shown in Fig. 8 of Appendix D.

3.2 Experiments on real data

We consider the experimental framework established in [32, 33] for the study of task similarity in relation to catastrophic forgetting. We use the Fashion-MNIST dataset [37] to generate upstream and downstream tasks: The upstream dataset—$\mathcal{D}_1 = \{\boldsymbol{x}_i^{(1)}, y_i^{(1)}\}_i$—consists in a pair of classes from the standard dataset, while the downstream dataset is generated by a linear interpolation of the upstream dataset with a second auxiliary dataset—$\tilde{\mathcal{D}} = \{\tilde{\boldsymbol{x}}_i, \tilde{y}_i\}_i$—containing a new pair of classes,

$$\mathcal{D}_2 = \{\boldsymbol{x}_i^{(2)}, y_i^{(2)}\}_i = \{\gamma \boldsymbol{x}_i^{(1)} + (1-\gamma)\tilde{\boldsymbol{x}}_i, \gamma y_i^{(1)} + (1-\gamma)\tilde{y}_i\}_i \tag{9}$$

where the parameter γ controls the task similarity. We then train a standard two-layer feedforward ReLU neural network on the two datasets using online SGD on a squared error loss. We consider a

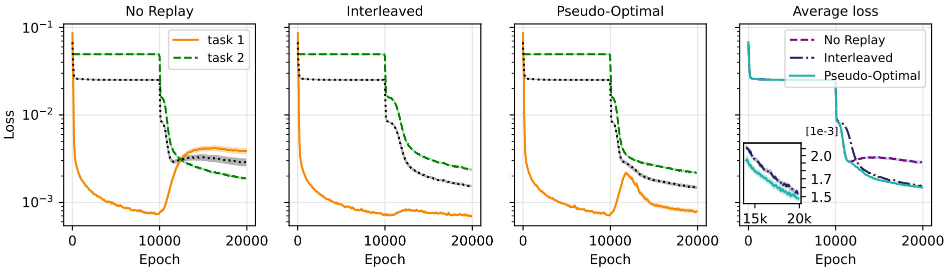

Figure 4: Training curves on the modified fashion MNIST task at similarity $\gamma = 0.5$. The network is trained for 10.000 epochs on the first task before switching to the second task and being trained for additional 10.000 epochs. The results are obtained from 100 realisations of the problem. The first three panels show the test loss on task 1 (solid orange), task 2 (dashed green), and their average (dotted black) for three training strategies, from left to right: no-replay, interleaved, and pseudo-optimal. The rightmost panel shows the average loss over the entire training.

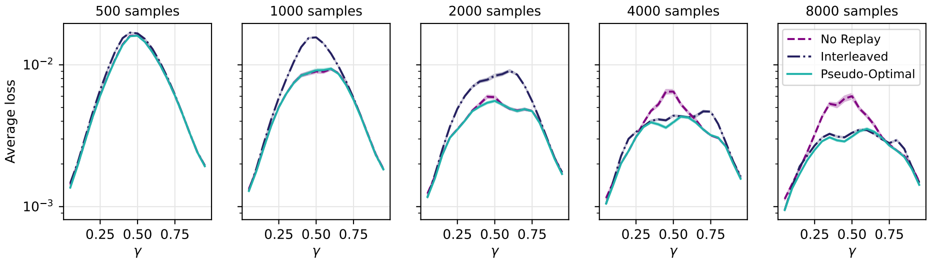

Figure 5: **Average loss comparison.** The figure focuses on the average loss and shows the final loss achieved by the three strategies as we increase the size of task 2 (from left to right: 500, 1.000, 2.000, 4.000, and 8.000 samples) while task 1 has always 10.000 samples. Individual panels show the performance of the three strategies as we span the value of γ form 0.05 to 0.95.

dynamical architecture [17, 18] where the readout weights are changed switching from one task to another, but the hidden layer is shared. During training, we apply the three strategies discussed in the previous sections: a no-replay strategy, a strategy with interleaved replay, and a "pseudo-optimal" strategy. Recall that the latter is inspired by the optimal protocol derived in the previous section. It consists of an initial phase of training exclusively on the new task until performance on both tasks becomes comparable, followed by a phase of interleaved replay. Crucially, this protocol can be easily implemented in practice, as it only requires an estimate of the generalisation error on the two tasks, which can be obtained in real-world settings.

Fig. 4 shows the training loss under the different training protocols for $\gamma = 0.5$. While the no-replay strategy appears to be successful for small downstream datasets (i.e., a few epochs in the online framework) in the longer run it leads to strong forgetting and high average loss. The interleaved is beneficial in the long run but largely slows down learning of the new task. Overall, the pseudo-optimal protocol identified in Sec. 3.1 shows a better performance over the entire trajectory.

This result is not limited to the specific value of γ. In Fig. 5, we show snapshots of the average loss for different downstream task sizes while spanning over a range of γs. This figure provides additional support to the observation reported previously that the no-replay strategy is optimal for small downstream tasks, the interleaved strategy is convenient for large downstream tasks, and the pseudo-optimal one combines the benefits of the two leading to the best performance overall. In summary, the pseudo-optimal strategy derived for the synthetic model performs well on real-world data. Notably, despite the differences between the synthetic and real settings—such as data structure—the pseudo-optimal strategy remains effective and robust across problems.

4 Discussion

Conclusion. In this work, we introduce a systematic approach for identifying and interpreting optimal task-selection strategies in synthetic learning settings. We consider a teacher-student scenario as a prototypical continual learning problem to achieve analytic understanding of supervised multi-task learning. We incorporate prior results on exact ODEs for high-dimensional online SGD dynamics into a control-theory framework that allows us to derive exact equations for the optimal protocols. Our theory reveals that optimal task-selection protocols are typically highly structured—alternating between focused learning and interleaved replay phases—and display a nontrivial interplay with task similarity. We also identify highly structured optimal learning rate schedules that synchronise with optimal task-selection to enhance overall performance. Finally, leveraging insights from the synthetic setting, we extract a pseudo-optimal strategy applicable to real tasks.

Limitations and Perspectives. This work takes a first step toward understanding the theory behind optimal training protocols for neural networks. In the following, we discuss current limitations and outline promising directions for future research. First, Pontryagin's maximum principle provides a necessary condition for optimality but does not guarantee a global optimum. Nevertheless, the strategies derived from this approach performed significantly better than previously proposed heuristics. Additionally, Pontryagin's principle does not easily extend to stochastic problems. This limitation is overcome in the high-dimensional limit where concentration results provide deterministic dynamical equations. For simplicity, we focus on i.i.d. Gaussian inputs, but our analysis can be extended to more structured data models [38, 39, 40] to study how the input distribution affects optimal task selection. In particular, we do not model the relative task difficulty—an important extension that naturally connects to the theory of curriculum learning [41, 42, 43, 44]. Furthermore, it would be interesting to go beyond the study of online dynamics to understand the impact of memorisation in batch learning settings [45]. Existing results in the spurious correlations [46] and fairness [47] literature suggest a strong dependence of the classifier's bias on the presentation order in batch learning. Our method can be applied to mean-field models—like [48, 49]—to theoretically investigate this phenomenon. An interesting extension of our work involves applying recently-developed statistical physics methods to the study of deeper networks and more complex architectures [50, 51, 52, 53]. Another interesting direction concerns finding optimal protocols for shaping, where task order significantly impacts both animal learning and neural networks [54, 55, 56].

References

[1] R Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Citeseer, 1993.

[2] Richard A Caruana. Multitask connectionist learning. In *Proceedings of the 1993 connectionist models summer school*, pages 372–379. Psychology Press, 1994.

[3] Rich Caruana. Learning many related tasks at the same time with backpropagation. *Advances in neural information processing systems*, 7, 1994.

[4] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

[5] Steven C Suddarth and YL Kergosien. Rule-injection hints as a means of improving network performance and learning time. In *European association for signal processing workshop*, pages 120–129. Springer, 1990.

[6] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[7] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[8] Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. Compete to compute. *Advances in neural information processing systems*, 26, 2013.

[9] Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[11] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[12] Robert M French. Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. In *Proceedings of the 13th annual cognitive science society conference*, volume 1, pages 173–178, 1991.

[13] Robert M French. Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, 4(3-4):365–377, 1992.

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[15] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[17] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Artificial intelligence and statistics*, pages 1453–1461. PMLR, 2012.

[18] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[19] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[20] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[21] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[22] Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 international joint conference on neural networks (IJCNN)*, pages 526–533. IEEE, 2017.

[23] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[24] David Saad and Sara Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. *Advances in neural information processing systems*, 8, 1995.

[25] David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995.

[26] Michael Biehl and Holm Schwarze. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and general*, 28(3):643, 1995.

[27] AA Feldbaum. On the synthesis of optimal systems with the aid of phase space. *Avtomatika i Telemehanika*, 16(2):129–149, 1955.

[28] LS Pontryagin. Some mathematical problems arising in connection with the theory of optimal automatic control systems. In *Proc. Conf. on Basic Problems in Automatic Control and Regulation*, 1957.

[29] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.

[30] Andreas Engel. *Statistical mechanics of learning*. Cambridge University Press, 2001.

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

[32] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

[33] Sebastian Lee, Stefano Sarao Mannelli, Claudia Clopath, Sebastian Goldt, and Andrew Saxe. Maslow's hammer in catastrophic forgetting: Node re-use vs. node activation. In *International Conference on Machine Learning*, pages 12455–12477. PMLR, 2022.

[34] Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983, 1989.

[35] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

[36] David Saad and Magnus Rattray. Globally optimal parameters for on-line learning in multilayer neural networks. *Physical review letters*, 79(13):2578, 1997.

[37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[38] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

[39] Bruno Loureiro, Cedric Gerbelot, Hugo Cui, Sebastian Goldt, Florent Krzakala, Marc Mezard, and Lenka Zdeborová. Learning curves of generic features maps for realistic datasets with a teacher-student model. *Advances in Neural Information Processing Systems*, 34:18137–18151, 2021.

[40] Urte Adomaityte, Gabriele Sicuro, and Pierpaolo Vivo. Classification of superstatistical features in high dimensions. In *2023 Conference on Neural Information Procecessing Systems*, 2023.

[41] Daphna Weinshall and Dan Amir. Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research*, 21(222):1–19, 2020.

[42] 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.

[43] Elisabetta Cornacchia and Elchanan Mossel. A mathematical model for curriculum learning for parities. In *International Conference on Machine Learning*, pages 6402–6423. PMLR, 2023.

[44] Emmanuel Abbe, Elisabetta Cornacchia, and Aryo Lotfi. Provable advantage of curriculum learning on parity targets with mixed inputs. *Advances in Neural Information Processing Systems*, 36:24291–24321, 2023.

[45] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[46] Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao. Procrustean training for imbalanced deep learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 92–102, 2021.

[47] Prakhar Ganesh, Hongyan Chang, Martin Strobel, and Reza Shokri. On the impact of machine learning randomness on group fairness. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1789–1800, 2023.

[48] Stefano Sarao Mannelli, Federica Gerace, Negar Rostamzadeh, and Luca Saglietti. Bias-inducing geometries: exactly solvable data model with fairness implications. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024.

[49] Anchit Jain, Rozhin Nobahari, Aristide Baratin, and Stefano Sarao Mannelli. Bias in motion: Theoretical insights into the dynamics of bias in sgd training. *arXiv preprint arXiv:2405.18296*, 2024.

[50] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[51] Blake Bordelon and Cengiz Pehlevan. The influence of learning rule on representation dynamics in wide neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.

[52] Riccardo Rende, Federica Gerace, Alessandro Laio, and Sebastian Goldt. Mapping of attention mechanisms to a generalized potts model. *Physical Review Research*, 6(2):023057, 2024.

[53] Lorenzo Tiberi, Francesca Mignacco, Kazuki Irie, and Haim Sompolinsky. Dissecting the interplay of attention paths in a statistical mechanics theory of transformers. *arXiv preprint arXiv:2405.15926*, 2024.

[54] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1938.

[55] William L Tong, Anisha Iyer, Venkatesh N Murthy, and Gautam Reddy. Adaptive algorithms for shaping behavior. *bioRxiv*, 2023.

[56] 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.

[57] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

[58] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[59] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

[60] Lei Zhang and Xinbo Gao. Transfer adaptation learning: A decade survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[61] Oussama Dhifallah and Yue M Lu. Phase transitions in transfer learning for high-dimensional perceptrons. *Entropy*, 23(4):400, 2021.

[62] Federica Gerace, Luca Saglietti, Stefano Sarao Mannelli, Andrew Saxe, and Lenka Zdeborová. Probing transfer learning with a model of synthetic correlated datasets. *Machine Learning: Science and Technology*, 3(1):015030, 2022.

[63] Alessandro Ingrosso, Rosalba Pacelli, Pietro Rotondo, and Federica Gerace. Statistical mechanics of transfer learning in fully-connected networks in the proportional limit. *arXiv preprint arXiv:2407.07168*, 2024.

[64] Haozhe Shan, Qianyi Li, and Haim Sompolinsky. Order parameters and phase transitions of continual learning in deep neural networks. *arXiv preprint arXiv:2407.10315*, 2024.

[65] Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in neural information processing systems*, 32, 2019.

[66] Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: the dynamics of learning with feedback alignment. In *International Conference on Machine Learning*, pages 8925–8935. PMLR, 2021.

[67] Stefano Sarao Mannelli, Yaraslau Ivashynka, Andrew M Saxe, Luca Saglietti, et al. Tilting the odds at the lottery: the interplay of overparameterisation and curricula in neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[68] Magnus Rattray and David Saad. Analysis of on-line training with optimal learning rates. *Physical Review E*, 58(5):6379, 1998.

[69] E Schlösser, D Saad, and M Biehl. Optimization of on-line principal component analysis. *Journal of Physics A: Mathematical and General*, 32(22):4061, 1999.

[70] David Saad and Magnus Rattray. Learning with regularizers in multilayer neural networks. *Physical Review E*, 57(2):2170, 1998.

[71] Magnus Rattray and David Saad. Globally optimal on-line learning rules for multi-layer neural networks. *Journal of Physics A: Mathematical and General*, 30(22):L771, 1997.

[72] Rodrigo Carrasco-Davis, Javier Masís, and Andrew M Saxe. Meta-learning strategies through value maximization in neural networks. *arXiv preprint arXiv:2310.19919*, 2023.

[73] Pierfrancesco Urbani. Disordered high-dimensional optimal control. *Journal of Physics A: Mathematical and Theoretical*, 54(32):324001, 2021.

[74] Jiequn Han, Qianxiao Li, et al. A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6(1):1–41, 2019.

[75] Xinyi Chen and Elad Hazan. Online control for meta-optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

A Related theoretical works

On the theoretical side, [57] pioneered the research on continual learning by deriving PAC bounds. More recently, further performance bounds have been obtained in the context of multi-task learning, few-shot learning, domain adaptation, and hypothesis transfer learning [58, 59, 60]. However, these results focused on worst-case analysis, offering bounds that may not reflect the typical performance of algorithms. In contrast, [61] began investigating the typical-case scenario, providing a precise characterisation of transfer learning in simple neural network models. [62, 63] extended this analysis to more complex architectures and generative models, allowing for a better description of the relation between tasks. Finally, [31, 33] proposed a theoretical framework for the study of the dynamics of continual learning with a focus on catastrophic forgetting. Their work provided a theoretical explanation for the surprising empirical results of [32], which revealed a non-monotonic relation between forgetting and task similarity, where maximal forgetting occurs at intermediate task similarity. Analogously, [64] studied a Gibbs formulation of continual learning in deep linear networks, and demonstrated how the interplay between task similarity and network architecture influences forgetting and knowledge transfer.

In recent years, several theoretical works on online learning dynamics in one-hidden-layer neural networks have addressed a range of machine learning problems, including over-parameterisation [65], algorithmic analysis [66**?**], and learning strategies [31, 42, 67]. However, these studies have not explored the problem from an optimal control perspective.

Early works addressed the optimality of hyperparameters in high-dimensional online learning for committee machines via control theory. These studies focused on optimising the learning rate [36, 68, 69], the regularisation [70], and the learning rule [71]. However, to the best of our knowledge, the problem of optimal task selection has not been explored yet. [72] and [**?**] applied optimal control to the dynamics of connectivist models of behaviour, but their analysis was limited to low-dimensional settings. [73] extended the Bellman equation to high-dimensional mean-field dynamical systems, though without considering learning processes.

Several other works have combined ideas from machine learning and optimal control. Notably, [74] interpreted deep learning as an optimal control problem on a dynamical system, where the control variables correspond to the network parameters. [75] formulated meta-optimization as an optimal control problem, but their analysis did not involve dimensionality reduction techniques nor did it address task selection.

B Details on the theoretical derivations

In this appendix, we provide detailed derivations of the equations in Sec. 2 of the main text. In the interest of completeness, we also report the derivation of the ODEs describing online SGD dynamics and the generalisation error as a function of the order parameters, first derived in [31]. We remind that inputs are $N-$dimensional vectors $\boldsymbol{x} \in \mathbb{R}^N$ with independent identically distributed (i.i.d.) standard Gaussian entries $x_i \sim \mathcal{N}(0,1)$, while the labels are generated by single-layer teacher networks: $y^{(t)} = g_*(\boldsymbol{x} \cdot \boldsymbol{w}_*^{(t)}/\sqrt{N})$, $t = 1, \dots, T$, with a different teacher for each task. The student is a one-hidden layer network that outputs the prediction:

$$\hat{y}^{(t)} = \sum_{k=1}^{K} v_k^{(t)} g\left(\frac{\boldsymbol{x} \cdot \boldsymbol{w}_k}{\sqrt{N}}\right) . \tag{10}$$

We focus on the *online* (on *one-pass*) setting, so that at each training step the student network is presented with a fresh example $\boldsymbol{x}^\mu$, $\mu = 1, \dots, P$, and $P/N \sim \mathcal{O}_N(1)$. The weights of the student are updated through gradient descent on $\frac{1}{2}(\hat{y}^{(t)} - y^{(t)})^2$ following the task-selection protocol t_c:

$$\begin{aligned} \boldsymbol{w}_k^{\mu+1} &= \boldsymbol{w}_k^{\mu} - \eta^\mu \Delta^{(t_c)\mu} \, v_k^{(t_c)\mu} \, g'\left(\lambda_k^\mu\right) \frac{\boldsymbol{x}^\mu}{\sqrt{N}} \, , \\ v_k^{(t)\mu+1} &= v_k^{(t)\mu} - \frac{\eta^\mu}{N} \Delta^{(t)\mu} \, g(\lambda_k^\mu) \delta_{t,t_c} \, , \\ \Delta^{(t)\mu} &:= \hat{y}^{(t)\mu} - y^{(t)\mu} = \sum_{k=1}^{K} v_k^{(t)} g(\lambda_k^\mu) - g_*(\lambda_*^{(t)\mu}) \, , \end{aligned} \tag{11}$$

where η^μ denotes the (possibly time-dependent) learning rate and we have rescaled it by N in the dynamics of the readout weights for future convenience. We have defined the preactivations, a.k.a. *local fields*,

$$\lambda_k^\mu := \frac{\boldsymbol{x}^\mu \cdot \boldsymbol{w}_k^\mu}{\sqrt{N}} \, , \qquad \lambda_*^{(t)\mu} := \frac{\boldsymbol{x}^\mu \cdot \boldsymbol{w}_*^{(t)}}{\sqrt{N}} \, . \tag{12}$$

Notice that, due to the online-learning setup, at each training epoch the input $\boldsymbol{x}$ is independent of the weights. Therefore, due to the Gaussianity of the inputs, the local fields are also jointly Gaussian with zero mean and second moments given by the *overlaps*:

$$\begin{aligned} M_{kt} &:= \mathbb{E}_{\boldsymbol{x}}\left[\lambda_k \lambda_*^{(t)}\right] = \frac{\boldsymbol{w}_k \cdot \boldsymbol{w}_*^{(t)}}{N} \, , \\ Q_{kh} &:= \mathbb{E}_{\boldsymbol{x}}\left[\lambda_k \lambda_h\right] = \frac{\boldsymbol{w}_k \cdot \boldsymbol{w}_h}{N} \, , \\ S_{tt'} &:= \mathbb{E}_{\boldsymbol{x}}\left[\lambda_*^{(t)} \lambda_*^{(t')}\right] = \frac{\boldsymbol{w}_*^{(t')} \cdot \boldsymbol{w}_*^{(t)}}{N} \, . \end{aligned} \tag{13}$$

B.1 Generalisation error as a function of the order parameters

We can write the generalisation error (Eq. 2 of the main text) as an average over the local fields:

$$\begin{aligned} \varepsilon_t\left(\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{W}_*\right) = \frac{1}{2} \sum_{k,h} v_k^{(t)} v_h^{(t)} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[g(\lambda_k) g(\lambda_h)\right] + \frac{1}{2} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[g_*(\lambda_*^{(t)})^2\right] \\ - \sum_{k} v_k^{(t)} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[g(\lambda_k) g_*(\lambda_*^{(t)})\right] \, . \end{aligned} \tag{14}$$

where the expectation is computed over the multivariate Gaussian distribution

$$
\begin{aligned}
P(\boldsymbol{\lambda}, \boldsymbol{\lambda}_*) &= \frac{1}{\sqrt{(2\pi)^{K+T}|\boldsymbol{C}|}} \exp\left(-\frac{1}{2}(\boldsymbol{\lambda}, \boldsymbol{\lambda}_*)^\top \boldsymbol{C}^{-1}(\boldsymbol{\lambda}, \boldsymbol{\lambda}_*)\right) \ , \\
\boldsymbol{C} &= \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{M} \\ \boldsymbol{M}^\top & \boldsymbol{S} \end{pmatrix} \ .
\end{aligned} \tag{15}
$$

From now on, we adopt the unified notation

$$
I_2(\beta, \rho) := \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[g_\beta(\lambda_\beta) g_\rho(\lambda_\rho)\right] \ , \tag{16}
$$

where β, ρ can refer both to the indices of the student weights k, h or the tasks t, t'. We can then rewrite the generalisation error as

$$
\varepsilon_t\left(\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{W}_*\right) = \frac{1}{2}\sum_{k,h} v_k^{(t)} v_h^{(t)} I_2(k,h) + \frac{1}{2} I_2(t,t) - \sum_k v_k^{(t)} I_2(k,t) \ . \tag{17}
$$

In all the results presented in Sec. 3, we consider $g(z) = g_*(z) = \operatorname{erf}\left(z/\sqrt{2}\right)$. In this case, there is an analytic expression for the integral I_2 [24]:

$$
I_2(\beta, \rho) = \frac{2}{\pi} \arcsin \frac{q_{\beta\rho}}{\sqrt{1+q_{\beta\beta}}\sqrt{1+q_{\rho\rho}}} \ , \tag{18}
$$

and we use the symbol q to denote generically an overlap from Eq. 13, according to the choice of indices β, ρ, e.g., $q_{kh} = Q_{kh}$, $q_{kt} = M_{kt}$, and $q_{tt_c} = S_{tt_c}$. In this special case, the generalisation error can be written explicitly as a function of the overlaps

$$
\begin{aligned}
\varepsilon_t\left(\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{W}_*\right) = \frac{1}{2\pi}\sum_{k,h} v_k^{(t)} v_h^{(t)} \arcsin \frac{Q_{kh}}{\sqrt{1+Q_{kk}}\sqrt{1+Q_{hh}}} + \frac{1}{2\pi} \arcsin \frac{S_{tt}}{1+S_{tt}} \\
- \frac{1}{\pi}\sum_k v_k^{(t)} \arcsin \frac{M_{kt}}{\sqrt{1+Q_{kk}}\sqrt{1+S_{tt}}} \ .
\end{aligned} \tag{19}
$$

B.2 Ordinary differential equations for the forward training dynamics

Given that the generalisation error depends only on the overlaps, in order to characterise the learning curves we need to compute the equations of motion for the overlaps from the SGD dynamics of the weights given in Eq. 11. The order parameter $S_{tt'}$ associated to the teachers is constant in time. We obtain an ODE for M_{kt} by multiplying both sides of the first of Eq. 11 by $\boldsymbol{w}_*^{(t)}$ and dividing by N:

$$
\frac{\boldsymbol{w}_k^{\mu+1} \cdot \boldsymbol{w}_*^{(t)}}{N} - \frac{\boldsymbol{w}_k^{\mu} \cdot \boldsymbol{w}_*^{(t)}}{N} = -\frac{\eta^\mu}{N} \Delta^{(t_c)\mu} v_k^{(t_c)\mu} g'(\lambda_k^\mu) \lambda_*^{(t)\mu} \ , \tag{20}
$$

where we stress the difference between t_c, the task selected for training at epoch μ, and t, the task for which we compute the overlap. We define a "training time" $\alpha = \mu/N$ and take the infinite-dimensional limit $N \to \infty$. The parameter α becomes continuous and M_{kt} concentrates to the solution of the following ODE:

$$
\frac{\mathrm{d}M_{kt}}{\mathrm{d}\alpha} = -\eta v_k^{(t_c)} \, \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[\Delta^{(t_c)} g'(\lambda_k) \lambda_*^{(t)}\right] := f_{\boldsymbol{M}, kt} \ , \tag{21}
$$

where the expectation is computed over the distribution in Eq. 15. The ODE for Q_{kh} is obtained similarly from Eq. 11:

$$
\begin{aligned}
\frac{\boldsymbol{w}_k^{\mu+1} \cdot \boldsymbol{w}_h^{\mu+1}}{N} - \frac{\boldsymbol{w}_k^{\mu} \cdot \boldsymbol{w}_h^{\mu}}{N} = -\frac{\eta^\mu}{N} \Delta^{(t_c)\mu} v_k^{(t_c)\mu} g'(\lambda_k^\mu) \lambda_h^\mu - \frac{\eta^\mu}{N} \Delta^{(t_c)\mu} v_h^{(t_c)\mu} g'(\lambda_h^\mu) \lambda_k^\mu \\
+ (\eta^\mu)^2 \left(\Delta^{(t_c)\mu}\right)^2 v_k^{(t_c)\mu} v_h^{(t_c)\mu} g'(\lambda_k^\mu) g'(\lambda_h^\mu) \frac{\boldsymbol{x} \cdot \boldsymbol{x}}{N} \ .
\end{aligned} \tag{22}
$$

In the infinite-dimensional limit, we find

$$
\frac{\mathrm{d}Q_{kh}}{\mathrm{d}\alpha} = -\eta v_k^{(t_c)} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[\Delta^{(t_c)} g'(\lambda_k^\mu) \lambda_h^\mu\right] - \eta v_h^{(t_c)} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[\Delta^{(t_c)} g'(\lambda_h^\mu) \lambda_k^\mu\right] \tag{23}
$$

$$
+ \eta^2 v_k^{(t_c)} v_h^{(t_c)} \mathbb{E}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}_*}\left[\left(\Delta^{(t_c)}\right)^2 g'(\lambda_k) g'(\lambda_h)\right] := f_{\boldsymbol{Q}, kh} \ . \tag{24}
$$

Finally, taking the infinite dimensional limit of the second Eq. 11, we find the ODE for the readout:

$$\frac{\mathrm{d}v_k^{(t)}}{\mathrm{d}\alpha} = -\eta\,\mathbb{E}_{\boldsymbol{\lambda},\boldsymbol{\lambda}_*}\left[\Delta^{(t)} g(\lambda_k)\right]\delta_{t,t_c} \coloneqq f_{\boldsymbol{V},tk}\ . \tag{25}$$

It is useful to write this system of ODEs in a more compact form. With the shorthand notation $\mathbb{Q} = (\mathrm{vec}(\boldsymbol{Q}), \mathrm{vec}(\boldsymbol{M}), \mathrm{vec}(\boldsymbol{V}))^\top$, $f_{\mathbb{Q}} = (\mathrm{vec}(f_{\boldsymbol{Q}}), \mathrm{vec}(f_{\boldsymbol{M}}), \mathrm{vec}(f_{\boldsymbol{V}}))^\top$, we can write

$$\frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} = f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), \boldsymbol{u}(\alpha)\right)\ , \qquad\qquad \alpha \in (0, \alpha_F]\ . \tag{26}$$

The initial condition for $\mathbb{Q}(0)$ is chosen to reproduce the random initialisation of the SGD algorithm. In particular, the initial first-layer weights and readout weights are drawn i.i.d. from a normal distribution with variances of 10^{-3} and 10^{-2}, respectively.

It is useful to write explicit expressions for the integrals involved in $f_{\mathbb{Q}}$ [31]. First, expanding the terms in $\Delta^{(t)}$, we can write

$$\begin{aligned} f_{\boldsymbol{Q},kh} = &-\eta v_k^{(t_c)}\left[\sum_{n=1}^{K} v_n^{(t_c)} I_3(n,k,h) - I_3(t_c,k,h)\right] \\ &- \eta v_h^{(t_c)}\left[\sum_{n=1}^{K} v_n^{(t_c)} I_3(n,h,k) - I_3(t_c,h,k)\right] \\ &+ \eta^2 v_k^{(t_c)} v_h^{(t_c)}\left[\sum_{n,m=1}^{K} v_n^{(t_c)} v_m^{(t_c)} I_4(n,m,k,h) + I_4(t_c,t_c,k,h)\right. \\ &\left. -2\sum_{n=1}^{K} v_n^{(t_c)} I_4(n,t_c,k,h)\right] \end{aligned} , \tag{27}$$

$$f_{\boldsymbol{M},kt} = -\eta v_k^{(t_c)} \sum_{n=1}^{K} v_n^{(t_c)} I_3(n,k,t) + \eta v_k^{(t_c)} I_3(t_c,k,t)\ , \tag{28}$$

$$f_{\boldsymbol{V},tk} = \eta\left[-\sum_{n=1}^{K} v_n^{(t_c)} I_2(k,n) + I_2(k,t_c)\right]\delta_{t,t_c}\ . \tag{29}$$

Similarly as in Eq. 16, we adopt the unified notation for the integrals

$$\begin{aligned} I_3(\beta,\rho,\zeta) &\coloneqq \mathbb{E}_{\boldsymbol{\lambda},\boldsymbol{\lambda}_*}\left[\lambda_\beta g_\rho'(\lambda_\rho) g(\lambda_\zeta)\right]\ , \\ I_4(\beta,\rho,\zeta,\tau) &\coloneqq \mathbb{E}_{\boldsymbol{\lambda},\boldsymbol{\lambda}_*}\left[g_\beta(\lambda_\beta) g_\rho(\lambda_\rho) g_\zeta'(\lambda_\zeta) g_\tau'(\lambda_\tau)\right]\ , \end{aligned} \tag{30}$$

where β, ρ, ζ, τ can refer both to the indices of the student weights k, h, n, m or the tasks t, t_c. In the special case $g(z) = g_*(z) = \mathrm{erf}(z/\sqrt{2})$, the integrals have explicit expressions as a function of the overlaps

$$\begin{aligned} I_3(\beta,\rho,\zeta) &= \frac{2q_{\rho\zeta}(1+q_{\beta\beta}) - 2q_{\beta\rho}q_{\beta\zeta}}{\pi\sqrt{\Lambda_3}(1+q_{\beta\beta})}\ , \\ I_4(\beta,\rho,\zeta,\tau) &= \frac{4}{\pi^2\sqrt{\Lambda_4}}\arcsin\frac{\Lambda_0}{\sqrt{\Lambda_1\Lambda_2}}\ , \end{aligned} \tag{31}$$

the symbol q denotes generically an overlap from Eq. 13, and

$$\begin{aligned} \Lambda_0 &= \Lambda_4\, q_{\beta\rho} - q_{\beta\tau}\, q_{\rho\tau}\,(1+q_{\zeta\zeta}) - q_{\beta\zeta}\, q_{\rho\zeta}\,(1+q_{\tau\tau}) + q_{\zeta\tau}\, q_{\beta\zeta}\, q_{\rho\tau} + q_{\zeta\tau}\, q_{\rho\zeta}\, q_{\beta\tau}\ , \\ \Lambda_1 &= \Lambda_4\,(1+q_{\beta\beta}) - q_{\beta\tau}^2\,(1+q_{\zeta\zeta}) - q_{\beta\zeta}^2\,(1+q_{\tau\tau}) + 2q_{\zeta\tau} q_{\beta\zeta}\, q_{\beta\tau}\ , \\ \Lambda_2 &= \Lambda_4\,(1+q_{\rho\rho}) - q_{\rho\tau}^2\,(1+q_{\zeta\zeta}) - q_{\rho\zeta}^2\,(1+q_{\tau\tau}) + 2q_{\zeta\tau} q_{\rho\zeta} q_{\rho\tau}\ , \\ \Lambda_3 &= (1+q_{\beta\beta})(1+q_{\rho\rho}) - q_{\beta\rho}^2\ , \\ \Lambda_4 &= (1+q_{\zeta\zeta})\,(1+q_{\tau\tau}) - q_{\zeta\tau}^2\ . \end{aligned} \tag{32}$$

B.3 Informal derivation of Pontryagin maximum principle

Let us consider the augmented cost function

$$\mathcal{F}[\mathbb{Q},\hat{\mathbb{Q}},\boldsymbol{u}] = h\left(\mathbb{Q}(\alpha_F)\right) + \int_0^{\alpha_F} \mathrm{d}\alpha\; \hat{\mathbb{Q}}(\alpha)^\top \left[-\frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} + f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha),\boldsymbol{u}(\alpha)\right)\right] \,, \tag{33}$$

where the conjugate variables $\hat{\mathbb{Q}}(\alpha)$ act as Lagrange multipliers, enforcing the dynamics at time α. Setting to zero variations with respect to $\hat{\mathbb{Q}}(\alpha)$ results in the forward dynamics

$$\frac{\delta\mathcal{F}[\mathbb{Q},\hat{\mathbb{Q}},\boldsymbol{u}]}{\delta\hat{\mathbb{Q}}(\alpha)} = 0 \Rightarrow \frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} = f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha),\boldsymbol{u}(\alpha)\right) \,. \tag{34}$$

Integrating by parts, we find

$$\begin{aligned}\mathcal{F}[\mathbb{Q},\hat{\mathbb{Q}},\boldsymbol{u}] = h\left(\mathbb{Q}(\alpha_F)\right) &+ \int_0^{\alpha_F} \mathrm{d}\alpha\; \hat{\mathbb{Q}}(\alpha)^\top f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha),\boldsymbol{u}(\alpha)\right) + \int_0^{\alpha_F} \mathrm{d}\alpha \frac{\mathrm{d}\hat{\mathbb{Q}}(\alpha)}{\mathrm{d}\alpha}^\top \mathbb{Q}(\alpha) \\ &- \hat{\mathbb{Q}}(\alpha_F)\mathbb{Q}(\alpha_F) + \hat{\mathbb{Q}}(0)\mathbb{Q}(0) \,.\end{aligned} \tag{35}$$

Setting to zero variations with respect to $\mathbb{Q}(\alpha)$ for $0 < \alpha < \alpha_F$, we find the backward dynamics

$$-\frac{\mathrm{d}\hat{\mathbb{Q}}(\alpha)^\top}{\mathrm{d}\alpha} = \hat{\mathbb{Q}}(\alpha)^\top \nabla_{\mathbb{Q}} f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha),\boldsymbol{u}(\alpha)\right) \,, \tag{36}$$

while for $\alpha = \alpha_F$ we get the final condition

$$\hat{\mathbb{Q}}(\alpha_F) = \nabla_{\mathbb{Q}} h(\mathbb{Q}(\alpha_F)) \,. \tag{37}$$

Note that we do not consider variations with respect to $\mathbb{Q}(0)$ as this quantity is fixed by the initial condition $\mathbb{Q}(0) = \mathbb{Q}_0$. Finally, minimizing the cost function with respect to the control $\boldsymbol{u}$, we get the optimality condition in Eq. 8 of the main text.

B.4 Optimal control framework

To determine the optimal control, we iterate Eqs. 4, 7, and 8 of the main text until convergence [**?**]. Let us consider first the case where the control is the current task $t_c(\alpha)$, such that $t_c(\alpha) = t$ if the network is trained on task $t \in \{1,\ldots,T\}$ at training time α. For simplicity, we focus on the case $T = 2$, but the following discussion is easily generalised to any T. In particular, since here $u(\alpha) = t_c(\alpha)$ the evolution equation 4 can be written as

$$\frac{\mathrm{d}\mathbb{Q}(\alpha)}{\mathrm{d}\alpha} = f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), t_c(\alpha)\right) \,, \quad \mathbb{Q}(0) = \mathbb{Q}_0 \,. \tag{38}$$

Similarly, the backward dynamics reads

$$-\frac{\mathrm{d}\hat{\mathbb{Q}}(\alpha)^\top}{\mathrm{d}\alpha} = \hat{\mathbb{Q}}(\alpha)^\top \nabla_{\mathbb{Q}} f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), t_c(\alpha)\right) \,, \tag{39}$$

with final condition

$$\hat{\mathbb{Q}}(\alpha_F) = \frac{1}{2}\nabla_{\mathbb{Q}}\varepsilon_1(\mathbb{Q}(\alpha_F)) + \frac{1}{2}\nabla_{\mathbb{Q}}\varepsilon_2(\mathbb{Q}(\alpha_F)) \,. \tag{40}$$

The optimality equation 8 yields

$$t_c^*(\alpha) = \underset{t_c\in\{1,2\}}{\mathrm{argmin}} \left\{\hat{\mathbb{Q}}(\alpha)^\top f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), t_c(\alpha) = t_c\right)\right\} \,. \tag{41}$$

Therefore, we find the explicit formula for the optimal task protocol

$$t_c^*(\alpha) = \begin{cases} 1 & \text{if } \hat{\mathbb{Q}}(\alpha)^\top \left[f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), t_c(\alpha) = 2\right) - f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), t_c(\alpha) = 1\right)\right] > 0 \\ 2 & \text{otherwise.}\end{cases} \tag{42}$$

Then, we start from a guess for the control variable $t_c(\alpha)$. We integrate Eq. 38 forward, obtaining the trajectory $\mathbb{Q}(\alpha)$ for $\alpha \in (0,\alpha_F)$. Then, we integrate the backward equation 39, starting from the final condition 40, obtaining the trajectory $\hat{\mathbb{Q}}(\alpha)$ for $\alpha \in (0,\alpha_F)$. Then, the control variable can be updated using Eq. 42 and used in the next iteration of the algorithm. These equations 38, 39, and 42 are iterated until convergence.

We next consider the joint optimisation of the learning rate schedule $\eta(\alpha)$ and the task protocol $t_c(\alpha)$. The optimality condition 8 can be written as

$$(t_c^*(\alpha), \eta(\alpha)) = \underset{t_c \in \{1,2\}, \eta \in \mathbb{R}^+}{\operatorname{argmin}} \left\{ \hat{\mathbb{Q}}(\alpha)^\top f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), (t_c(\alpha), \eta(\alpha)) = (t_c, \eta)\right) \right\} . \tag{43}$$

Crucially, the function $\hat{\mathbb{Q}}^\top f_{\mathbb{Q}}(\mathbb{Q}, (t_c, \eta))$ turns out to be quadratic in η. Explicitly,

$$\hat{\mathbb{Q}}^\top f_{\mathbb{Q}}(\mathbb{Q}, (t_c, \eta)) = a\eta^2 + b\eta \,, \tag{44}$$

where

$$\begin{aligned} a = \sum_{k,h=1}^{K} \hat{Q}_{kh} v_k^{(t_c)} v_h^{(t_c)} \Bigg[& \sum_{n,m=1}^{K} v_n^{(t_c)} v_m^{(t_c)} I_4(n,m,k,h) + I_4(t_c,t_c,k,h) \\ & -2 \sum_{n=1}^{K} v_n^{(t_c)} I_4(n,t_c,k,h) \Bigg] \,, \end{aligned} \tag{45}$$

and

$$\begin{aligned} b = - \sum_{k,h=1}^{K} \hat{Q}_{kh} \Bigg\{ & v_k^{(t_c)} \left[\sum_{n=1}^{K} v_n^{(t_c)} I_3(n,k,h) - I_3(t_c,k,h) \right] \\ & + v_h^{(t_c)} \left[\sum_{n=1}^{K} v_n^{(t_c)} I_3(n,h,k) - I_3(t_c,h,k) \right] \Bigg\} \\ & - \sum_{k=1}^{K} \sum_{t=1}^{T} \hat{M}_{kt} \left[v_k^{(t_c)} \sum_{n=1}^{K} v_n^{(t_c)} I_3(n,k,t) - v_k^{(t_c)} I_3(t_c,k,t) \right] \\ & + \sum_{k=1}^{K} \hat{v}_k^{(t_c)} \left[- \sum_{n=1}^{K} v_n^{(t_c)} I_2(k,n) + I_2(k,t_c) \right] . \end{aligned} \tag{46}$$

Performing the minimization over η first, we obtain

$$\eta^*(\alpha, t_c) = -\frac{b}{2a} \,. \tag{47}$$

The minimisation over t_c yields

$$t_c^*(\alpha) = \begin{cases} 1 & \text{if } \hat{\mathbb{Q}}(\alpha)^\top \left[f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), (1, \eta^*(\alpha,1))\right) - f_{\mathbb{Q}}\left(\mathbb{Q}(\alpha), (2, \eta^*(\alpha,2))\right) \right] > 0 \\ 2 & \text{otherwise.} \end{cases} \tag{48}$$

and hence

$$\eta^*(\alpha) = \eta^*(\alpha, t_c^*(\alpha)) \,. \tag{49}$$

Interestingly, we observe that the learning rate schedule has a different functional form depending on the current task t_c. This can be seen in Fig. 3 where the learning rate switches between two different schedules depending on the current task t_c.

C Readout layer convergence properties

In this appendix, we examine the asymptotic behaviour of the readout layer weights during the late stages of training. Once the two hidden neurons have specialised—each aligning with one of the teacher vectors—we expect the readout weights corresponding to the incorrect teacher to be suppressed. Specifically, if $\boldsymbol{w}_1 = \boldsymbol{w}_*^{(1)}$ and $\boldsymbol{w}_2 = \boldsymbol{w}_*^{(2)}$, the learning dynamics should drive the readout weights $\boldsymbol{v}^{(1)} = (v_1^{(1)}, v_2^{(1)})^\top$ and $\boldsymbol{v}^{(2)} = (v_1^{(2)}, v_2^{(2)})^\top$ towards $\boldsymbol{v}^{(1)} = (1,0)^\top$ and $\boldsymbol{v}^{(2)} = (0,1)^\top$, representing full recovery of the teacher network. As shown in Fig. 7 of Appendix D, the time required to suppress the off-diagonal weights $v_2^{(1)}$ and $v_1^{(2)}$ increases as $\gamma \to 1$. This is intuitive, as higher task similarity γ reduces the distinction between tasks, slowing the suppression of the off-diagonal weights. In what follows, we derive analytically the convergence timescale $\alpha_{\rm conv}$ of the readout layer as a function of the task similarity γ and the learning rate η. As in the main text, we consider the case $K = T = 2$. From the overlap trajectories in Fig. 7 for $\gamma > 0.3$, we observe that the cosine similarity quickly approaches unity, i.e., $|M_{kt}|/\sqrt{Q_{kk}} \approx \delta_{kt}$, which corresponds to perfect feature recovery. Therefore, the decrease in performance for $\gamma > 0.3$ seen in Fig. 2 must be attributed to the dynamics of the second layer. Indeed, in Fig. 7, we observe a slowdown in the readout dynamics as $\gamma \to 1$.

Assuming perfect convergence of the feature layer to $\boldsymbol{w}_1 = \boldsymbol{w}_*^{(1)}$ and $\boldsymbol{w}_2 = \boldsymbol{w}_*^{(2)}$, we consider the dynamics of the readout layer while training on task $t = 1$. We expect the corresponding readout layer to converge to the specialised configuration $\boldsymbol{v}^{(1)} = (v_1^{(1)}, v_2^{(1)}) = (1, 0)^\top$ and we would like to compute the convergence rate as a function of γ. The dynamics of the readout layer reads

$$\begin{aligned}\frac{{\rm d}v_1^{(1)}}{{\rm d}\alpha} &= \eta \left[\frac{1}{3}(1 - v_1^{(1)}) - \frac{2}{\pi} \arcsin\left(\frac{\gamma}{2}\right) v_2^{(1)}\right] , \qquad (50)\\ \frac{{\rm d}v_2^{(1)}}{{\rm d}\alpha} &= \eta \left[\frac{2}{\pi} \arcsin\left(\frac{\gamma}{2}\right)(1 - v_1^{(1)}) - \frac{1}{3} v_2^{(1)}\right] ,\end{aligned}$$

which can be rewritten as

$$\frac{\rm d}{{\rm d}\alpha} \begin{pmatrix} 1 - v_1^{(1)} \\ v_2^{(1)} \end{pmatrix} = \eta \boldsymbol{A} \begin{pmatrix} 1 - v_1^{(1)} \\ v_2^{(1)} \end{pmatrix} , \qquad (51)$$

where

$$\boldsymbol{A} = \begin{bmatrix} -1/3 & a \\ a & -1/3 \end{bmatrix} , \qquad (52)$$

and $a = 2 \arcsin\left(\gamma/2\right)/\pi$. Note that $a < 1/3$ for $0 < \gamma < 1$, hence $\boldsymbol{A}$ is negative definite, implying convergence to $\boldsymbol{v}^{(1)} = (1, 0)^\top$. The rate of convergence is determined by the smallest eigenvalue (in absolute value): $a - 1/3$. The associated convergence timescale is therefore

$$\alpha_{\rm conv} = \frac{3\pi}{\eta(\pi - 6 \arcsin\left(\gamma/2\right))} . \qquad (53)$$

This timescale is a monotonically increasing function of γ and diverges as $\gamma \to 1$ with $\alpha_{\rm conv} \approx \sqrt{3}\pi/(2\eta(1 - \gamma))$. This result explains the performance decrease of the optimal strategy as $\gamma \to 1$. In summary, the performance decrease for $\gamma \to 0$ is due to the first-layer weights, while for $\gamma \to 1$ it is related to the readout weights.

D Supplementary figures

Fig. 7 describes the dynamics of the optimal replay strategy for different values of task similarity in the same setting as Fig. 2 of the main text. In particular, the upper panel displays the evolution of the magnitude of the readout weights $|v_k^{(t)}|$, while the lower panel shows the trajectory of the cosine similarity $|M_{kt}|/\sqrt{Q_{kk}}$.

Fig. 8 compares the values of the loss at the end of training, averaged on both tasks, for different task-selection strategies. In particular, it highlights the performance gap between the four replay strategies at constant learning rate considered in the main text (no-replay, interleaved, optimal and pseudo-optimal) and the strategy that simultaneously optimise over task-selection and learning rate.

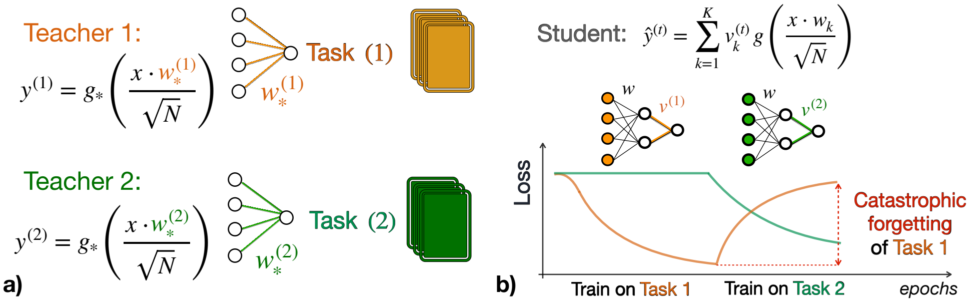
Teacher 1:
$y^{(1)} = g_*\left(\frac{x \cdot w_*^{(1)}}{\sqrt{N}}\right)$
Task (1)
$w_*^{(1)}$
Teacher 2:
$y^{(2)} = g_*\left(\frac{x \cdot w_*^{(2)}}{\sqrt{N}}\right)$
Task (2)
$w_*^{(2)}$
a)
Student: $\hat{y}^{(t)} = \sum_{k=1}^{K} v_k^{(t)} g\left(\frac{x \cdot w_k}{\sqrt{N}}\right)$
w
$v^{(1)}$
w
$v^{(2)}$
Loss
Catastrophic forgetting of Task 1
Train on Task 1
Train on Task 2
epochs
b)

Figure 6: **Pictorial representation of the continual learning task in the teacher-student setting.** A "student" network is trained on i.i.d. inputs from two teacher networks, defining two different tasks (panel **a**). The student has sufficient capacity to learn both tasks. However, sequential training results in catastrophic forgetting, where the performance on a previously learned task significantly deteriorates when a new task is introduced (panel **b**). Parameters: $K = T = 2$.

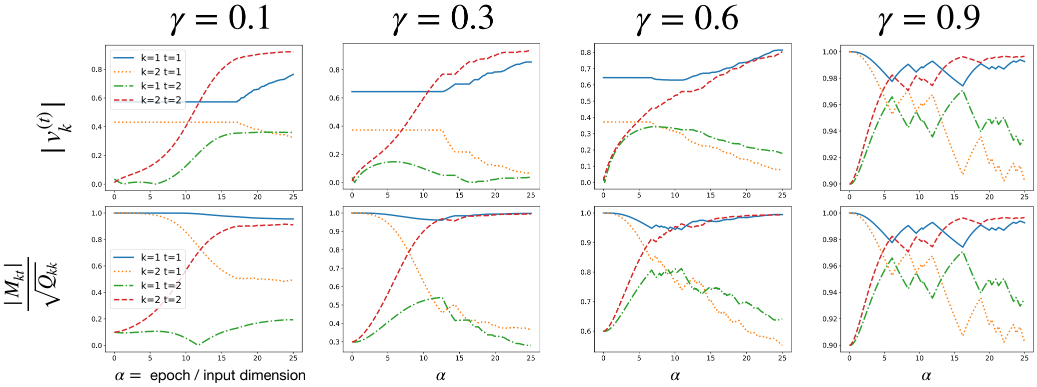

Figure 7: **Overlap dynamics with optimal replay.** We plot the absolute value of the task-dependent readout weights $|v_k^{(t)}|$ (upper panel) and the cosine similarity $|M_{kt}|/\sqrt{Q_{kk}}$ as a function of the training time α. Different columns refer to different choices of task similarity $\gamma = 0.1, 0.3, 0.6, 0.9$.

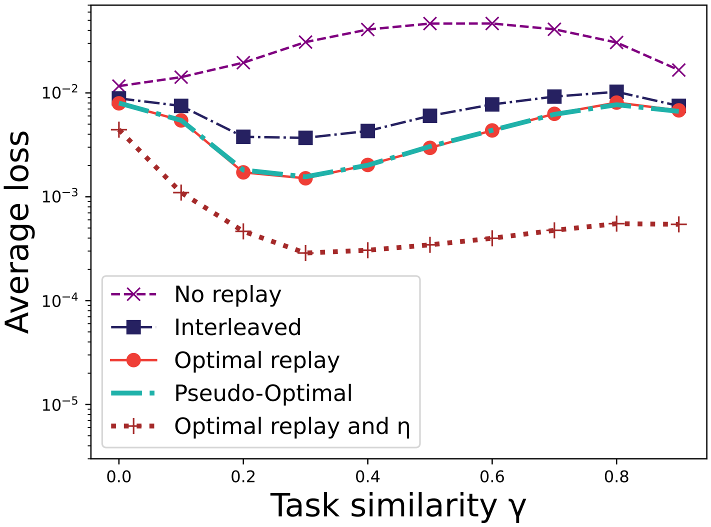

Figure 8: **Adopting an optimal learning rate schedule leads to major perfomance improvement.** Average loss on both tasks at the end of the second training phase as a function of task similarity γ under the same setting and parameters as Fig. 2 of the main text. The top four lines correspond to different strategies at constant learning rate $\eta = 1$: no replay (purple crosses), optimal replay (red dots), interleaved (blue squares), pseudo-optimal replay (cyan dashed line). The bottom curve (brown plus signs) corresponds to jointly optimal replay and learning rate schedules (see Fig. 3).