A DYNAMICAL THEORY OF LORA IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the widespread use of Low-Rank Adaptation (LoRA), little is known about its dynamics in continual learning and the mechanisms by which low-rank updates affect catastrophic forgetting. We provide an asymptotically exact dynamical characterization of LoRA in a solvable two-task teacher-student model. In the high-dimensional online-learning limit, we derive a closed system of ordinary differential equations for a finite set of macroscopic order parameters, yielding exact expressions for the generalization errors throughout both the initial Task 1 learning phase and the subsequent LoRA fine-tuning on Task 2. The theory quantitatively matches finite-dimensional simulations and exposes two characteristic effects of LoRA: low-rank adaptation reduces interference with features learned on the first task, but its initialization slows adaptation to the second task. Building on this mechanistic picture, we analyze a state-dependent masking strategy that freezes hidden units carrying the strongest first-task representations and restricts adaptation to the complementary subspace. This structural partitioning markedly reduces forgetting, while preserving plasticity on the new task. Our framework further clarifies the role of adapter rank: transfer improves only up to the intrinsic dimensionality of the target task and saturates beyond it, while forgetting continues to grow with rank. These results provide a dynamical and geometric account of how low-rank adaptation organizes information across sequential tasks and are qualitatively reproduced on a sequential MNIST benchmark.

1 INTRODUCTION

Modern machine-learning systems are increasingly adapted to new tasks rather than trained from scratch. This paradigm is particularly important for large pre-trained models, for which updating all parameters can be computationally and memory intensive. Parameter-efficient fine-tuning (PEFT) methods address this problem by restricting adaptation to a small set of trainable parameters (Xu et al., 2023; Han et al., 2024). Among them, Low-Rank Adaptation (LoRA) (Hu et al., 2021) has become a widely used approach: the pre-trained weights are frozen and adaptation is performed through a trainable low-rank perturbation.

Freezing the pre-trained weights, however, does not by itself guarantee that the behavior learned before fine-tuning is preserved. The LoRA update changes the effective representation seen by the network and can therefore interfere with features that were useful for previous tasks. This issue is particularly relevant in continual learning (Wang et al., 2023), where models are trained sequentially and must acquire new information without catastrophically forgetting previously learned tasks (McCloskey & Cohen, 1989). Recent methods have consequently sought to control the subspace in which LoRA updates occur, for example by constructing directions that reduce interference with previous tasks (Liang & Li, 2024) or by adapting underutilized spectral directions (Rüdiger & Raschka, 2026). Yet understanding *why* low-rank adaptation retains or forgets information requires a dynamical description of how the adapter interacts with the representation learned before the task switch.

A growing theoretical literature has begun to characterize different aspects of LoRA, including its expressivity, convergence, initialization, and optimization dynamics (Zeng & Lee, 2024; Xu et al., 2023; Kim et al., 2025). Existing dynamical analyses either condition on a fixed pretrained state (Nwemadji et al., 2026) or do not resolve the time dependence across the two stages (Duranthon et al., 2026). These works provide important insights into LoRA adaptation, but leave open

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.

a complementary question central to continual learning: *how does a low-rank update dynamically reorganize representations learned on a previous task, and how does this geometry jointly shape transfer to the new task and forgetting of the old one?*

We address this question in a solvable two-task teacher-student model, building on the high-dimensional online-learning framework of Lee et al. (2021; 2022). Task 1 is learned by standard SGD; at the task switch, the learned representation is frozen and Task 2 is learned through the LoRA update. In the high-dimensional limit, we derive a closed system of deterministic ordinary differential equations (ODEs) for a finite set of macroscopic overlaps that determine the generalization errors on both tasks. LoRA changes the dynamical equations after the switch: the pretrained overlaps become fixed, while additional order parameters track the geometry of the LoRA adapter relative to the frozen representation and to both teachers. This framework reveals reduced interference but slower adaptation under LoRA, and allows us to study subspace restriction, adapter rank, and task similarity within a common stability-plasticity picture.

Main contributions.

- **Dynamical theory of sequential LoRA.** We derive an asymptotically exact high-dimensional description of Task 1 feature learning followed by Task 2 LoRA adaptation. The stochastic dynamics close onto a finite system of ODEs that jointly determine transfer, forgetting, and representation geometry.
- **LoRA-specific mechanisms and structured adaptation.** We show that LoRA freezes the pretrained overlaps while introducing new adapter-representation and adapter-teacher overlaps, revealing reduced interference but slower adaptation from the initialization (Biderman et al., 2024). Motivated by existing subspace-constrained adaptation methods (Liang & Li, 2024; Rüdiger & Raschka, 2026), we incorporate a state-dependent masking rule into the same dynamical theory, showing that it strongly reduces forgetting while preserving Task 2 performance; we observe the same qualitative behavior on MNIST.
- **Rank, similarity, and stability-plasticity.** We characterize how adapter rank and task similarity control transfer and forgetting: useful transfer saturates once the adapter can represent the target feature space, while forgetting increases. Interference between tasks peaks at intermediate task similarity and is strongly suppressed by structured masking, mitigating forgetting.

2 Related Work

High-dimensional learning dynamics. Our analysis builds on a rigorous statistical-physics description of online learning (Goldt et al., 2019), where high-dimensional stochastic updates reduce to deterministic dynamics for a finite set of macroscopic order parameters (Gardner & Derrida, 1989; Saad & Solla, 1995a;b; Biehl & Schwarze, 1995). The generalization error can then be expressed in terms of this sufficient statistics across training time. Lee et al. (2021) extended this formalism to continual learning under standard SGD, showing how task similarity controls transfer and forgetting; subsequent work studied feature re-use and optimal control (Lee et al., 2022; Mori et al., 2025). Our Task 1 dynamics follow this framework, but after the switch we freeze the learned representation and optimize a factorized low-rank perturbation, which requires a different macroscopic closure.

Continual learning with LoRA. Continual-learning methods mitigate catastrophic forgetting through regularization, replay, or architectural separation (Kirkpatrick et al., 2016; Chaudhry et al., 2019; Rusu et al., 2016). Recent LoRA-based approaches instead constrain the adapter update (Lu et al., 2025; Wei et al., 2024; Liang & Li, 2024; Che et al., 2026). Most relevant here, InfLoRA selects directions designed to reduce previous-task interference (Liang & Li, 2024), while MiCA restricts adaptation using the spectral structure of the pretrained model (Rüdiger & Raschka, 2026). Our objective is complementary: rather than proposing a new subspace-selection principle, we use a solvable dynamical model to analyze how constraining the LoRA update relative to previously learned features affects the stability–plasticity trade-off.

Theoretical analyses of LoRA. LoRA theory has addressed expressivity, optimization, convergence, and generalization (Zeng & Lee, 2024; Xu et al., 2025; Kim et al., 2025; Kratsios et al.,

2025). Most closely related, Nwemadji et al. (2026) study LoRA fine-tuning dynamics conditional on a pretrained state. Our setting instead tracks the full sequential process: we dynamically generate the pretrained state through Task 1 learning and subsequently track both Task 2 acquisition and Task 1 forgetting. Duranthon et al. (2026) instead provide a high-dimensional asymptotic theory of pretraining and LoRA fine-tuning in a solvable attention model, without resolving the time-dependent training dynamics across the two stages. Our framework therefore connects dynamical continual-learning theory with a time-resolved theory of low-rank adaptation.

3 Continual Learning with LoRA: Problem Setting

We study continual learning in a two-task teacher-student setting (Gardner & Derrida, 1989; Lee et al., 2021). Let

$$\phi(\boldsymbol{\xi}; \boldsymbol{J}, \mathbf{v}) = \sum_{i=1}^{D} v_i \, g\left(\frac{\boldsymbol{J}_i \boldsymbol{\xi}}{\sqrt{N}}\right) \tag{1}$$

denote a two-layer fully connected network with D hidden units, first-layer weights $\boldsymbol{J} \in \mathbb{R}^{D\times N}$, readout weights $\boldsymbol{v} \in \mathbb{R}^{D}$, and activation function g. The notation $f(x;\, y)$ distinguishes the variable x from the fixed parameter y. Inputs are sampled independently as $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$.

Teachers and task similarity. The two tasks are generated by fixed teacher networks indexed by $* \in \{\dagger, \ddagger\}$, with parameters $(\boldsymbol{W}^*, \boldsymbol{v}^*)$, where $\boldsymbol{W}^* \in \mathbb{R}^{M\times N}$ and $\boldsymbol{v}^* \in \mathbb{R}^{M}$. Their targets are

$$y^* = \phi(\boldsymbol{\xi}; \boldsymbol{W}^*, \boldsymbol{v}^*). \tag{2}$$

Teacher $\dagger$ defines Task 1 and teacher $\ddagger$ Task 2. To control task similarity, we draw $W^{\dagger}_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and set $\boldsymbol{W}^{\ddagger} = c\,\boldsymbol{W}^{\dagger} + \sqrt{1-c^2}\,\mathbf{Z},\ \mathbf{Z}_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$, with $\mathbf{Z}$ independent of $\boldsymbol{W}^{\dagger}$ and $c \in [0,1]$. In the high-dimensional limit $N \to \infty$ with $M = O(1)$,

$$\frac{1}{N}\boldsymbol{W}^{\dagger}(\boldsymbol{W}^{\ddagger})^T \longrightarrow c\,\boldsymbol{I}_M. \tag{3}$$

Hence, $c = 0$ corresponds to asymptotically orthogonal teacher features, whereas $c = 1$ gives identical first-layer teacher representations. For both teachers, we take uniform readout weights with opposite signs, $\boldsymbol{v}^{\dagger}_i = +1$ and $\boldsymbol{v}^{\ddagger}_i = -1$ for $i \in [M]$.

Student and sequential learning protocol. The student has shared first-layer weights $\boldsymbol{J} \in \mathbb{R}^{K\times N}$ and task-specific readout heads $\boldsymbol{h}^{\dagger}, \boldsymbol{h}^{\ddagger} \in \mathbb{R}^{K}$, with prediction

$$\hat{y}^* = \phi(\boldsymbol{\xi}; \boldsymbol{J}, \boldsymbol{h}^*). \tag{4}$$

Task identity is therefore known at evaluation time, and since $\boldsymbol{h}^{\dagger}$ is frozen after the switch (below), the forgetting we measure is caused by changes in the shared first layer alone. Unless stated otherwise, we consider $K = 2M$, so that the student has sufficient hidden-layer capacity to represent both teachers without an intrinsic width bottleneck.

Training proceeds sequentially. During Task 1, $\boldsymbol{J}$ and $\boldsymbol{h}^{\dagger}$ are optimized by online SGD on the squared loss using samples generated by teacher $\dagger$. Writing μ for the number of examples presented, the macroscopic dynamics evolve on the rescaled time $\tau = \frac{\mu}{N}$, and each task is trained for $\tau \in [0, \alpha]$; α therefore sets the number of training steps per task, $T = \alpha N$ (see Appendix B). At the task switch, examples begin to be generated by teacher $\ddagger$. Let $\boldsymbol{J}_s$ denote the first-layer weights at the task switch. During Task 2, $\boldsymbol{J}_s$ and $\boldsymbol{h}^{\dagger}$ are frozen, while adaptation is performed through a LoRA update

$$\boldsymbol{J} = \boldsymbol{J}_s + \Delta\boldsymbol{J}, \qquad \Delta\boldsymbol{J} = \frac{\gamma}{\sqrt{L}}\boldsymbol{B}\boldsymbol{A}, \tag{5}$$

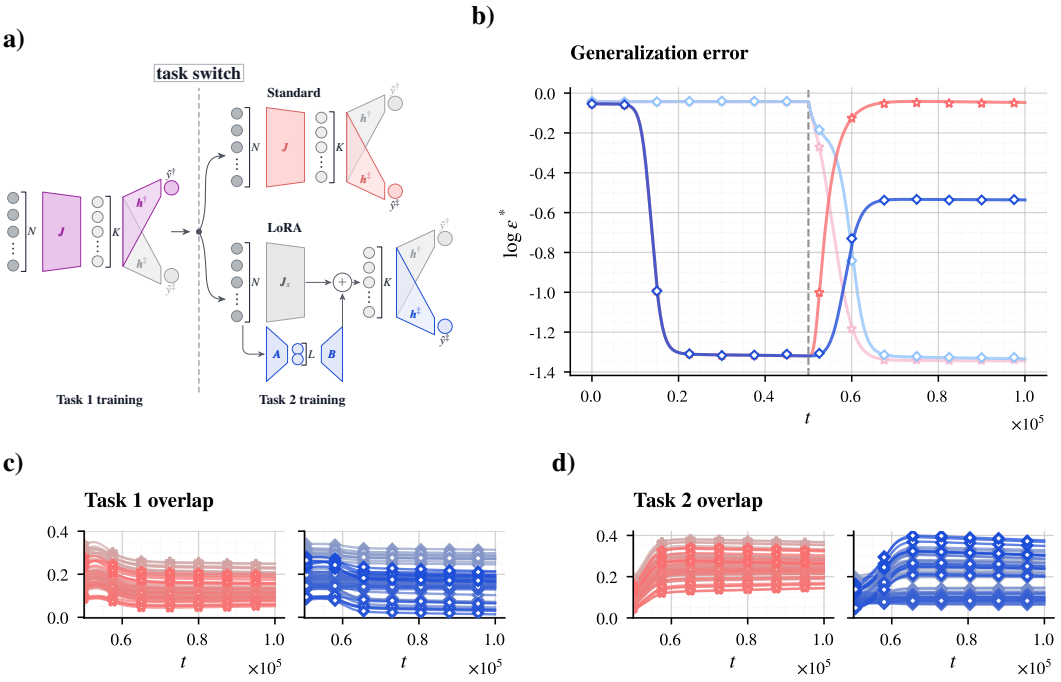

Figure 1: **Generalization and representation dynamics under LoRA and full fine-tuning for a sequential training.** *a)* Schematic of the sequential training setting. We first train the first-layer weights and a readout head on a synthetic dataset corresponding to Task 1. We then train on a correlated dataset corresponding to Task 2, where we either update the first-layer weights or freeze them and instead train a low-rank adapter. In both cases, we train a second readout head. *b)* Task 1 (dark) and Task 2 (light) generalization errors for full fine-tuning and LoRA. The gray dashed line marks the task switch. *c–d)* Student overlaps with teachers as defined in Eqs. (18)-(19) after the task switch for both full fine-tuning (red) and LoRA (blue). Solid lines denote theoretical ODE predictions and markers finite-dimensional simulations. Parameters: $N = 10^3$, $K = 10$, $M = 5$, $L = 5$, $c = 0.5$, $\alpha = 50$.

where $\boldsymbol{A} \in \mathbb{R}^{L\times N}$, $\boldsymbol{B} \in \mathbb{R}^{K\times L}$, L is the adapter rank, and γ controls the update scale. During Task 2, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{h}^\ddagger$ are trainable. The LoRA factors are initialized so that $\Delta \boldsymbol{J} = \boldsymbol{0}$ at the task switch, ensuring that inserting the adapter does not immediately perturb the Task 1 representation. Details on initialization of the low-rank adapters are reported in Appendix F.

Generalization error and high-dimensional limit. Performance on task $* \in \{\dagger, \ddagger\}$ is measured by the population generalization error

$$\epsilon^* = \frac{1}{2}\mathbb{E}_{\boldsymbol{\xi}}\left[\left(\phi(\boldsymbol{\xi}; \boldsymbol{J}, \boldsymbol{h}^*) - \phi(\boldsymbol{\xi}; \boldsymbol{W}^*, \boldsymbol{v}^*)\right)^2\right]. \tag{6}$$

We work in the online-learning regime, where each SGD step uses an independent sample from the data-generating distribution. Throughout we take $\eta_{\boldsymbol{J}}$,$\eta_{\boldsymbol{A}}$, $\eta_{\boldsymbol{B}}$ and $\eta_{\boldsymbol{h}}$ for the learning rates of the first layer, the two adapter factors and the readouts; their N-scalings are fixed in Appendix C and their values are reported in Appendix H. In the following, we show that, in the limit $N \to \infty$ with $K, M, L = O(1)$, the stochastic training dynamics concentrate onto a closed deterministic system for a finite set of macroscopic order parameters, from which we can track the evolution of the generalization error of both tasks with training time.

4 A Dynamical Theory of LoRA in Continual Learning

We now derive a closed macroscopic description of the sequential learning dynamics in the high-dimensional limit for activation function $g(z) = \mathrm{erf}(z/\sqrt{2})$. Our analysis builds on the online

teacher-student framework of Lee et al. (2021; 2022). The key difference arises after the task switch: instead of continuing to update the student first-layer weights, we freeze the Task 1 representation and optimize a factorized low-rank perturbation. This changes both the microscopic dynamics and the set of macroscopic quantities required to obtain a closed theory.

Task 1: standard feature learning. During training on Task 1, the student weights $(\boldsymbol{J}^\mu, (\boldsymbol{h}^\dagger)^\mu)$ are updated via online SGD on the squared loss. The resulting prediction error on the μ-th example is

$$\Delta^{\dagger,\mu} = \sum_{k=1}^{K} (h_k^\dagger)^\mu g(x_k^\mu) - \sum_{m=1}^{M} v_m^\dagger g(\rho_m^\mu). \tag{7}$$

where, for an input $\boldsymbol{\xi}^\mu$, ρ_m^μ and x_k^μ define the Task 1 teacher and student preactivations

$$\rho_m^\mu = \frac{\boldsymbol{W}_m^\dagger \boldsymbol{\xi}^\mu}{\sqrt{N}}, \qquad x_k^\mu = \frac{\boldsymbol{J}_k^\mu \boldsymbol{\xi}^\mu}{\sqrt{N}}. \tag{8}$$

The scaling in N is chosen such that the macroscopic quantities evolve on the time scale $\tau = \mu/N$ as $N \to \infty$, this phase coincides with the standard continual-learning dynamics of Lee et al. (2021). All details on these existing results can be found in Appendix B.

Task 2: low-rank adaptation of a frozen representation. At the task switch, the feature matrix is frozen at $\boldsymbol{J}_s$ and the effective representation becomes

$$\boldsymbol{J} = \boldsymbol{J}_s + \frac{\gamma}{\sqrt{L}} \boldsymbol{B}\boldsymbol{A}. \tag{9}$$

During this phase, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{h}^\ddagger$ are updated by online SGD, while $\boldsymbol{J}_s$ and $\boldsymbol{h}^\dagger$ remain frozen. The resulting Task 2 prediction error is therefore

$$\Delta^\ddagger = \sum_{k=1}^{K} h_k^\ddagger g(\tilde{x}_k) - \sum_{p=1}^{M} v_p^\ddagger g(\nu_p). \tag{10}$$

which depends on the Task-2 teacher fields, the frozen student fields and the L LoRA fields

$$\nu_p^\mu = \frac{\boldsymbol{W}_p^\ddagger \boldsymbol{\xi}^\mu}{\sqrt{N}}, \qquad x_k = \frac{(\boldsymbol{J}_s)_k \boldsymbol{\xi}}{\sqrt{N}}, \qquad z_i = \frac{\boldsymbol{A}_i \boldsymbol{\xi}}{\sqrt{N}}, \tag{11}$$

so that the adapted student preactivation is

$$\tilde{x}_k = x_k + \frac{\gamma}{\sqrt{L}} \sum_{i=1}^{L} B_{ki} z_i. \tag{12}$$

Macroscopic order parameters. The population generalization errors in (6) depend on the N-dimensional input only through the scalar fields. Since $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{I}_N)$ and all fields above are linear functions of $\boldsymbol{\xi}$, they are jointly zero-mean Gaussian. Their distribution is therefore fully determined by their second moments, which are normalized inner products between the corresponding weight vectors. Before the task switch, we track

$$\boldsymbol{Q}_{kl} := \langle x_k x_l \rangle, \qquad \boldsymbol{R}_{km} := \langle x_k \rho_m \rangle, \qquad \boldsymbol{U}_{kp} := \langle x_k \nu_p \rangle, \tag{13}$$

together with the fixed teacher overlaps

$$\boldsymbol{T}_{mn} := \langle \rho_m \rho_n \rangle, \qquad \boldsymbol{V}_{mp} := \langle \rho_m \nu_p \rangle, \qquad \boldsymbol{S}_{pq} := \langle \nu_p \nu_q \rangle, \tag{14}$$

which, by the teacher construction satisfy $\boldsymbol{T} = \boldsymbol{S} = \boldsymbol{I}_M$ and $\boldsymbol{V} = c\,\boldsymbol{I}_M$ in the high-dimensional limit.

Equivalently, by replacing the field definition

$$\boldsymbol{Q}_{kl} = \frac{1}{N}\boldsymbol{J}_k\boldsymbol{J}_l^T, \qquad \boldsymbol{R}_{km} = \frac{1}{N}\boldsymbol{J}_k(\boldsymbol{W}_m^\dagger)^T, \qquad \boldsymbol{U}_{kp} = \frac{1}{N}\boldsymbol{J}_k(\boldsymbol{W}_p^\ddagger)^T, \tag{15}$$

with analogous expressions for $\boldsymbol{T}, \boldsymbol{V}, \boldsymbol{S}$. Thus $\boldsymbol{Q}$ describes the geometry of the student representation, $\boldsymbol{R}$ and $\boldsymbol{U}$ its alignment with Tasks 1 and 2, and $\boldsymbol{T}, \boldsymbol{V}, \boldsymbol{S}$ the fixed geometry of the teachers.

LoRA introduces four additional overlap matrices,

$$\boldsymbol{\Phi}_{ij} := \langle z_i z_j \rangle, \qquad \boldsymbol{\Xi}_{ki} := \langle x_k z_i \rangle, \qquad \boldsymbol{\Lambda}_{mi} := \langle \rho_m z_i \rangle, \qquad \boldsymbol{\Gamma}_{pi} := \langle \nu_p z_i \rangle. \tag{16}$$

In weight space,

$$\boldsymbol{\Phi}_{ij} = \frac{1}{N}\boldsymbol{A}_i\boldsymbol{A}_j^T, \qquad \boldsymbol{\Xi}_{ki} = \frac{1}{N}(\boldsymbol{J}_s)_k\boldsymbol{A}_i^T, \qquad \boldsymbol{\Lambda}_{mi} = \frac{1}{N}\boldsymbol{W}_m^\dagger\boldsymbol{A}_i^T, \qquad \boldsymbol{\Gamma}_{pi} = \frac{1}{N}\boldsymbol{W}_p^\ddagger\boldsymbol{A}_i^T. \tag{17}$$

Hence $\boldsymbol{\Phi}$ describes the geometry of the trainable LoRA directions, $\boldsymbol{\Xi}$ their alignment with the frozen student representation, and $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ their alignment with the Task 2 and Task 1 teachers, respectively. Since $\boldsymbol{B} \in \mathbb{R}^{K\times L}$ remains finite-dimensional as $N \to \infty$, its entries are tracked explicitly.

How LoRA changes the dynamical closure. This is the central modification relative to standard continual learning. Under full fine-tuning, the student overlaps themselves continue to evolve after the task switch. Under LoRA, $\boldsymbol{Q}_s, \boldsymbol{R}_s, \boldsymbol{U}_s$ are fixed at their Task 1 values, and the geometry of the effective representation is reconstructed from these frozen overlaps and the adapter variables. In particular,

$$\langle \tilde{x}_k \rho_m \rangle = (\boldsymbol{R}_s)_{km} + \frac{\gamma}{\sqrt{L}}(\boldsymbol{B}\boldsymbol{\Lambda}^T)_{km}, \tag{18}$$

$$\langle \tilde{x}_k \nu_p \rangle = (\boldsymbol{U}_s)_{kp} + \frac{\gamma}{\sqrt{L}}(\boldsymbol{B}\boldsymbol{\Gamma}^T)_{kp}, \tag{19}$$

and

$$\langle \tilde{x}_k \tilde{x}_l \rangle = (\boldsymbol{Q}_s)_{kl} + \frac{\gamma}{\sqrt{L}}\left[(\boldsymbol{\Xi}\boldsymbol{B}^T)_{kl} + (\boldsymbol{B}\boldsymbol{\Xi}^T)_{kl}\right] + \frac{\gamma^2}{L}(\boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{B}^T)_{kl}. \tag{20}$$

These relations completely determine the covariance matrix of the adapted preactivations and therefore the population errors on both tasks. More precisely, in the high-dimensional limit, during Task 2 training the generalization errors of both tasks are functions of the order parameters

$$\begin{aligned} \lim_{N\to\infty} \epsilon^\dagger &= \epsilon^\dagger(\boldsymbol{\Xi}, \boldsymbol{\Phi}, \boldsymbol{\Lambda}, \boldsymbol{B};\ \boldsymbol{Q}_s, \boldsymbol{R}_s, \boldsymbol{T}, \boldsymbol{h}^\dagger, \boldsymbol{v}^\dagger) \\ \lim_{N\to\infty} \epsilon^\ddagger &= \epsilon^\ddagger(\boldsymbol{\Xi}, \boldsymbol{\Phi}, \boldsymbol{\Gamma}, \boldsymbol{B}, \boldsymbol{h}^\ddagger;\ \boldsymbol{Q}_s, \boldsymbol{U}_s, \boldsymbol{S}, \boldsymbol{v}^\ddagger) \end{aligned} \tag{21}$$

The corresponding explicit expressions are given in Appendix C.

Deterministic high-dimensional dynamics. The evolution equations follow by combining the microscopic SGD updates with the definitions above and taking the limit $N \to \infty$ at fixed $\tau = \mu/N$. As an illustration, consider

$$\boldsymbol{\Gamma}_{pi} = \frac{1}{N}\boldsymbol{W}_p^\ddagger\boldsymbol{A}_i^T, \tag{22}$$

which measures the alignment between the i-th LoRA direction and the p-th Task 2 teacher feature. Its evolution is

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_{pi}}{\mathrm{d}\tau} = -\eta_{\boldsymbol{A}}\frac{\gamma}{\sqrt{L}}\sum_{k=1}^{K}\boldsymbol{h}_k^\ddagger\boldsymbol{B}_{ki}\left\langle \Delta^\ddagger g'(\tilde{x}_k)\nu_p \right\rangle. \tag{23}$$

Analogous calculations yield a closed deterministic system for $\boldsymbol{\Phi}, \boldsymbol{\Xi}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{B}$ and $\boldsymbol{h}^\ddagger$. For $g(z) = \mathrm{erf}(z/\sqrt{2})$, all Gaussian expectations can be evaluated in closed form as algebraic functions of the instantaneous order parameters. The complete ODE system and the corresponding generalization-error expressions for Task 2 training are provided in Appendix C.

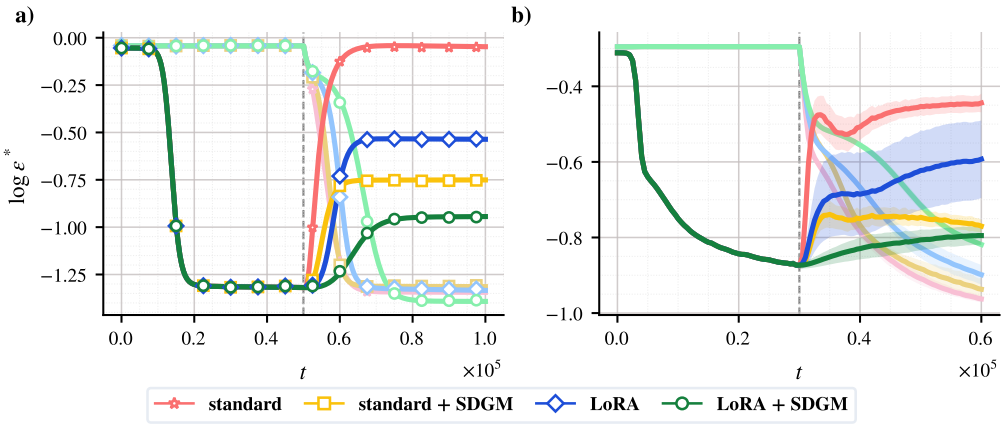

Figure 2: **State-dependent masking reduces forgetting.** *a)* Generalization dynamics for full fine-tuning, LoRA, and their SDGM-constrained variants. Parameters: $N = 10^3$, $K = 10$, $M = 5$, $L = 5$, $\kappa = 5$, $c = 0.5$, $\alpha = 50$. *b)* Corresponding results on a real experiment on the MNIST dataset sequential training (see Appendix G for details). SDGM improves Task 1 retention while preserving comparable Task 2 performance. For the experiment, the curves are averaged over 10 independent training realizations.

LoRA versus full fine-tuning. Figure 1b compares the resulting LoRA dynamics (blue) with standard full fine-tuning (red). Full fine-tuning rapidly increases the Task 1 error after the switch, whereas LoRA preserves substantially more of the previously learned representation while reaching a comparable asymptotic Task 2 error. The theoretical trajectories (solid lines) closely match finite-dimensional simulations (markers).

The overlap dynamics provide a geometric explanation. Under full fine-tuning, the shared representation itself moves toward Task 2, thereby modifying features acquired on Task 1. Under LoRA, the pretrained component remains fixed and the change in Task 1 and Task 2 alignment is mediated only by $\boldsymbol{B}\boldsymbol{\Lambda}^T$ and $\boldsymbol{B}\boldsymbol{\Gamma}^T$, respectively. As illustrated in Fig. 1c-d, LoRA increases alignment with the new task while preserving a larger fraction of the Task 1 alignment than full fine-tuning. This provides a direct representation-level explanation for its reduced forgetting.

In Fig. 1, the task switch occurs before the student fully specializes to the Task 1 teacher, as indicated by Fig. 1c. This regime is practically relevant, since training is typically stopped once a target performance is reached rather than after complete representational specialization; in addition, the time constant for symmetric-subspace escape grows with K, so full specialization at $K > M$ requires substantially longer training. Results in the fully specialized regime are reported in Appendix E.

LoRA also adapts more slowly at early times, as observed empirically (Liu et al., 2024; Li et al., 2025). Our theory attributes this to the dynamics of the adapter: with $\boldsymbol{A}$ initialized at zero, the Task-2 signal must first build up the adapter overlaps $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ through the rank-L bottleneck, while the up-projection $\boldsymbol{B}$ evolves on the slower readout timescale. This transient delays the growth of the Task 2 overlap, as seen in Fig. 1d, and slows early adaptation relative to full fine-tuning.

5 Stability and Plasticity through the Lens of the Theory

5.1 State-Dependent Masking Mitigates Forgetting

The theory suggests that forgetting can be reduced by preventing the LoRA update from acting on feature directions that are strongly used by Task 1. In the multi-head teacher-student model, the magnitude of the Task 1 readout coefficient $|\boldsymbol{h}_i^\dagger|$ provides a simple measure of the importance of hidden unit i for the first task. We therefore rank the hidden units by $|\boldsymbol{h}_i^\dagger|$ at the task switch, freeze the κ largest, and restrict Task 2 adaptation to the complementary set. We refer to this state-dependent partition as State-Dependent Gradient Masking (SDGM).

Within LoRA, we implement this constraint by fixing the up-projection factor to a sparse mask $\boldsymbol{\Omega} \in \{0,1\}^{K\times L}$ whose nonzero rows correspond only to the plastic hidden units, while optimizing the down-projection $\boldsymbol{A}$ and the Task 2 readout $\boldsymbol{h}^{\ddagger}$. The effective representation is therefore

$$\boldsymbol{J} = \boldsymbol{J}_s + \frac{\gamma}{\sqrt{L}}\boldsymbol{\Omega}\boldsymbol{A}. \tag{24}$$

Because $\boldsymbol{\Omega}$ is constructed from the Task 1 state and then held fixed, the same macroscopic theory applies by setting $\boldsymbol{B} = \boldsymbol{\Omega}$ and $d\boldsymbol{B}/d\tau = 0$, while evolving $\boldsymbol{\Phi}, \boldsymbol{\Xi}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}$ and $\boldsymbol{h}^{\ddagger}$. The explicit mask construction and the corresponding modification of the ODE system are given in Appendix D.1.

This construction is closely related to subspace-constrained PEFT methods such as InfLoRA and MiCA (Liang & Li, 2024; Rüdiger & Raschka, 2026). The distinction is that here the protected subspace is selected directly from the network state reached after Task 1, using the task-specific readout as an importance score, and its effect on the subsequent dynamics can be followed analytically.

Figure 2a shows that SDGM (green) substantially improves Task 1 retention relative to vanilla LoRA (blue) while preserving similar asymptotic Task 2 performance, with the theoretical trajectories closely matching finite-dimensional simulations. The inverse-selection control in Appendix D.2, which protects the least important Task 1 units instead, at the same number of trainable directions, produces substantially more forgetting, showing that the gain depends on *which* directions are protected rather than only on reducing the dimensionality of the trainable subspace. The same qualitative effect persists with ReLU activations as illustrated in Appendix D.3.

Applying the same state-dependent mask to full fine-tuning (yellow) also reduces forgetting, confirming that targeted protection of Task 1-relevant features is beneficial independently of the low-rank parameterization (Fig. 2). However, in the regime considered here, combining this restriction with LoRA (green) yields the strongest Task 1 retention at comparable Task 2 performance.

Figure 2b shows analogous behavior on a sequential MNIST. In this case, Task 1 is a binary classification problem distinguishing digits smaller than 5 from digits greater than or equal to 5, while Task 2 distinguishes even from odd digits. As we can see, SDGM plus LoRA again improves Task 1 retention while preserving competitive Task 2 performance; experimental details are in Appendix G.

5.2 Adapter Rank and the Stability–Plasticity Trade-off

The LoRA rank L controls the dimensionality of the trainable update and therefore provides a natural handle on the stability-plasticity trade-off. This dynamical framework allows us to quantify this trade-off by varying L while tracking both transfer to Task 2 and forgetting on Task 1, defined as

$$\text{Forgetting} = \log \epsilon^{\dagger}_{\text{final}} - \log \epsilon^{\dagger}_{s}, \qquad \text{Transfer} = \log \epsilon^{\ddagger}_{s} - \log \epsilon^{\ddagger}_{\text{final}}, \tag{25}$$

where $\epsilon^{*}_{\text{final}}$ is the error at the end of the sequential training for task $*$. Figure 3c shows that, for vanilla LoRA (blue), increasing L initially improves Task 2 transfer but also increases Task 1 forgetting. With SDGM (green), transfer likewise improves with rank, while forgetting remains substantially lower because adaptation is restricted away from Task 1-relevant directions. For this comparison, we set $\kappa = K - L$ so that the number of frozen directions decreases with the adapter capacity. At the same time, this choice allows the student to allocate exactly L units for Task 2.

A second feature is that transfer gains saturate as L approaches the teacher width M. Since Task 2 is generated by M independent feature directions, increasing the rank beyond this scale provides little additional representational benefit on Task 2, while retaining increasingly less information on Task 1, resulting in increased forgetting.

5.3 Task similarity and interference

We next vary the teacher similarity $c \in [0,1]$. As shown in Fig. 3a, forgetting under full fine-tuning (red) and vanilla LoRA (blue) is strongly non-monotonic, with maximal interference at intermediate similarity, consistent with previous continual-learning analyses (Ramasesh et al., 2020; Lee et al., 2021; 2022; Jarvis et al., 2025). When c is small, the tasks occupy nearly orthogonal feature directions and interact weakly; when c approaches one, previously learned features can be reused. At

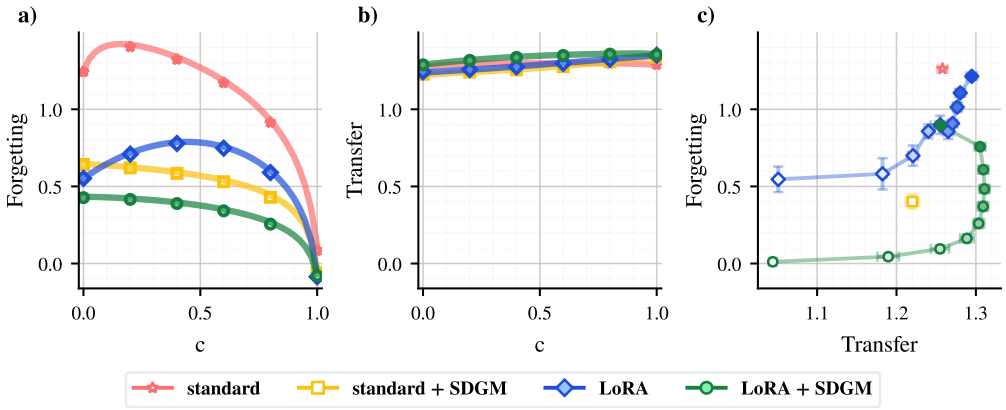

Figure 3: **Impact of LoRA on the Stability-Plasticity Trade-off.** *a)* Task 1 Forgetting and *b)* Task 2 Transfer as a function of teacher similarity c. Solid lines denote theoretical predictions and markers simulation. *c)* Forgetting versus Transfer as the LoRA rank L varies from light ($L = 1$) to dark marker ($L = 10$) for $c = 0.5$. Vanilla LoRA gains transfer at the cost of increased forgetting, whereas LoRA+SDGM maintains greater stability. Transfer gains saturate as L approaches the teacher width M. For SDGM, $\kappa = K - L$. We average over 10 different seeds. Lines in panel *c* are shown to guide the eyes. Parameters: $N = 10^3$, $K = 10$, $M = 5$, $\alpha = 50$. In panel *a* and *b* we used $L = 5$.

intermediate similarity, however, the tasks overlap enough to induce updates along shared directions while remaining sufficiently different to distort the Task 1 representation.

SDGM substantially suppresses this intermediate-similarity interference while preserving comparable Task 2 transfer across the range of c (panel b). This highlights the key geometric limitation of vanilla LoRA: restricting the *rank* of the update does not control its *orientation* relative to previously learned features. By protecting Task 1-relevant directions and redirecting adaptation toward the complementary subspace, SDGM improves the stability-plasticity trade-off especially when the update is already low rank.

6 Discussion and Conclusions

We developed a high-dimensional dynamical theory of LoRA in continual learning that follows the complete sequential process from Task 1 feature learning to Task 2 low-rank adaptation. The theory shows that LoRA changes both the geometry and timescale of learning: freezing the pretrained representation reduces interference, whereas the rank-restricted, zero-initialized adapter must first build up alignment through the low-rank bottleneck, which slows early adaptation. More generally, low rank alone does not prevent forgetting; the orientation of the adaptation subspace relative to previously learned features is equally important. This perspective explains why state-dependent masking reduces forgetting and clarifies how rank and task similarity shape the stability-plasticity trade-off.

Our analysis is deliberately restricted to a solvable two-layer, two-task online-learning model, so its quantitative predictions should not be transferred directly to large deep networks. Its purpose is instead to isolate mechanisms that are difficult to disentangle empirically. The qualitative agreement on sequential MNIST suggests that these mechanisms extend beyond the analytically tractable setting and motivates studying dynamically constrained adaptation in deeper networks and longer task sequences. Code to reproduce the experiments will be made publicly available.

AI use statement

The authors used large language models as general-purpose writing and coding assistants: for improving the phrasing and clarity of the manuscript, and for routine support during the implementation of the numerical experiments. All theoretical derivations were carried out by the authors,

all closed-form expressions were checked against finite-dimensional simulations, and all scientific claims, figures, and conclusions are the authors' own. No language model was used for ideation of the core contribution, for literature search, or as a source of results reported in the paper.

Ethics Statement

This work is a theoretical study of learning dynamics in a solvable teacher-student model. The synthetic data are generated from randomly sampled teacher networks and involve no human subjects, no personally identifiable information, and no sensitive attributes. The only real dataset used is MNIST, a standard public benchmark of handwritten digits. We see no direct pathway from this work to harmful applications; its potential societal impact is indirect, through a better understanding of when parameter-efficient fine-tuning preserves or degrades previously acquired behavior.

Reproducibility statement

All results in this paper are reproducible from the equations given in the appendices together with the accompanying code. The Gaussian integrals used throughout are stated in closed form in Appendix A. Appendix B gives the dynamics of standard sequential training, covering both Task 1 learning and full fine-tuning on Task 2; Appendix C gives the Task 2 LoRA dynamics, including the complete ODE system and both generalization errors. The SDGM mask construction and the modification it induces in the ODE system are given in Appendix D.1, the inverse-selection control in Appendix D.2, and the validation under ReLU activations in Appendix D.3 Results in the specialized regime are reported in Appendix E, and the initialization and scaling conventions for the LoRA factors in Appendix F Details of the sequential MNIST experiment are given in Appendix G. Code to reproduce every figure, both the ODE integration and the finite-dimensional simulations are available anonymously in this anonymous repository.

Author Contributions

Author contributions will be included in the camera-ready version.

Acknowledgments

Acknowledgments are omitted to preserve anonymity and will be included in the camera-ready version.

References

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. LoRA learns less and forgets less. *Trans. Mach. Learn. Res.*, 2024. ISSN 2835-8856. doi: 10.48550/arXiv.2405.09673. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Michael Biehl and H Schwarze. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and General*, 28(3):643–656, 01 1995. doi: 10.1088/0305-4470/28/3/018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019.

Chang Che, Ziqi Wang, Pengwan Yang, Cheems Wang, Hui Ma, and Zenglin Shi. Lora in lora: Towards parameter-efficient architecture expansion for continual visual instruction tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19978–19986, Mar. 2026. doi: 10.1609/aaai.v40i24.39082. URL https://ojs.aaai.org/index.php/AAAI/article/view/39082.

O. Duranthon, F. Boncoraglio, and L. Zdeborová. High-dimensional theory of lora fine-tuning in a solvable attention model, 2026. URL https://arxiv.org/abs/2606.05899.

E. Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983–1994, 1989. doi: 10.1088/0305-4470/22/12/004. URL https://hal.science/hal-03285594.

Sebastian Goldt, Madhu Advani, Andrew Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/cab070d53bd0d200746fb852a922064a-Paper.pdf.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2403.14608.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Devon Jarvis, Sebastian Lee, Clémentine Carla Juliette Dominé, Andrew M Saxe, and Stefano Sarao Mannelli. A theory of initialisation's impact on specialisation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RQz7szbVDs.

Junsu Kim, Jaeyeon Kim, and Ernest K. Ryu. LoRA training provably converges to a low-rank global minimum or it fails loudly (but it probably won't fail). In *International Conference on Machine Learning*, 2025. doi: 10.48550/arXiv.2502.09376. URL https://openreview.net/forum?id=o9zDYV4Ism.

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 of the United States of America*, 114(13):3521–3526, 2016. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Anastasis Kratsios, Tin Sum Cheng, Aurelien Lucchi, and Haitz Sáez de Ocáriz Borde. Sharp generalization bounds for foundation models with asymmetric randomized low-rank adapters, 2025. URL https://arxiv.org/abs/2506.14530.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

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.

Sebastian Lee, Stefano Sarao Mannelli, Claudia Clopath, Sebastian Goldt, and Andrew Saxe. Maslow's hammer for catastrophic forgetting: Node re-use vs node activation. In *arXiv.org*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12455–12477. PMLR, 2022. doi: 10.48550/arXiv.2205.09029.

Shiwei Li, Xiandi Luo, Xing Tang, Haozhao Wang, Hao Chen, Weihong Luo, Yuhua Li, Xiuqiang He, and Ruixuan Li. Beyond zero initialization: Investigating the impact of non-zero initialization on LoRA fine-tuning dynamics. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35519–35535. PMLR, 13–19 Jul 2025. doi: 10.48550/arXiv.2505.23194. URL https://proceedings.mlr.press/v267/li25bm.html.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Computer Vision and Pattern Recognition*, pp. 23638–23647. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02231.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.940.

Michael McCloskey and Neal J. Cohen. *Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem*, pp. 109–165. Elsevier, 1989. ISBN 9780125433242. doi: 10.1016/s0079-7421(08)60536-8. URL http://dx.doi.org/10.1016/S0079-7421(08)60536-8.

Francesco Mori, Stefano Sarao Mannelli, and Francesca Mignacco. Optimal protocols for continual learning via statistical physics and control theory. In *International Conference on Learning Representations*, volume 2025, pp. 59198–59220, 2025.

Gibbs Nwemadji, Bruno Loureiro, and Jean Barbier. When pre-training hurts lora fine-tuning: a dynamical analysis via single-index models, 2026. URL https://arxiv.org/abs/2602.02855.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=LhY8QdUGSuw.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2016. URL https://arxiv.org/abs/1606.04671.

Sten Rüdiger and Sebastian Raschka. Mica learns more knowledge than lora and full fine-tuning, 2026.

David Saad and Sara A. Solla. On-line learning in soft committee machines. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 52(4):4225–4243, Oct 1995a. doi: 10.1103/PhysRevE.52.4225. URL https://link.aps.org/doi/10.1103/PhysRevE.52.4225.

David Saad and Sara A. Solla. Exact solution for on-line learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337–4340, 1995b. ISSN 1079-7114. doi: 10.1103/physrevlett.74.4337. URL http://dx.doi.org/10.1103/PhysRevLett.74.4337.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2023. doi: 10.1109/TPAMI.2024.3367329.

Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *IEEE Workshop/Winter Conference on Applications of Computer Vision*, pp. 6634–6645. IEEE, 2024. doi: 10.1109/WACV61041.2025.00646.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023. URL https://arxiv.org/abs/2312.12148.

Ziqing Xu, Hancheng Min, Lachlan Ewen MacDonald, Jinqi Luo, Salma Tarmoun, Enrique Mallada, and Rene Vidal. Understanding the learning dynamics of lora: A gradient flow perspective on low-rank adaptation in matrix factorization. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4636–4644. PMLR, 03–05 May 2025. doi: 10.48550/arXiv.2503.06982. URL https://proceedings.mlr.press/v258/xu25h.html.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.17513. URL https://openreview.net/forum?id=likXVjmh3E.

A INTEGRALS COMPUTATION

To derive the closed-form equations for the various quantities we will determine, we make use of several quantities involving Gaussian integrals. For completeness, we report their expressions in this Appendix, following Saad & Solla (1995b).We denote by

$$\langle f(\boldsymbol{x})\rangle_{\boldsymbol{x}} \equiv \int d\boldsymbol{x}\, P(\boldsymbol{x}) f(\boldsymbol{x}), \tag{26}$$

the expectation of a function f with respect to its joint Gaussian distribution with zero mean and covariance matrix $\boldsymbol{C}$. All the computations that follow are specific to $g(x) = \mathrm{erf}(x/\sqrt{2})$.

For the two-dimensional case, we define

$$I_2 = \langle g(x) g(y)\rangle_{(x,y)}\,. \tag{27}$$

This Gaussian integral admits the closed-form expression

$$I_2(a,b) = \frac{2}{\pi}\arcsin\left(\frac{\boldsymbol{C}_{ab}}{\sqrt{\boldsymbol{C}_{aa}+1}\sqrt{\boldsymbol{C}_{bb}+1}}\right). \tag{28}$$

Likewise, the computation of a three-dimensional gaussian average

$$I_3 = \langle g'(x)\, y\, g(z)\rangle_{(x,y,z)} \tag{29}$$

can be computed to get

$$I_3(a,b,c) = \frac{2}{\pi}\frac{1}{\sqrt{\Lambda_3(a,c)}}\frac{\boldsymbol{C}_{bc}(\boldsymbol{C}_{aa}+1) - \boldsymbol{C}_{ab}\boldsymbol{C}_{ac}}{\boldsymbol{C}_{aa}+1}. \tag{30}$$

Finally, the four-dimensional gaussian integral

$$I_4 = \langle g'(x)\, g'(y)\, g(z)\, g(w)\rangle_{(x,y,z,w)} \tag{31}$$

results in

$$I_4(a,b,c,d) = \frac{4}{\pi^2\sqrt{\Lambda_4(a,b)}}\arcsin\left(\frac{\Lambda_0(a,b,c,d)}{\sqrt{\Lambda_1(a,b,c)}\sqrt{\Lambda_2(a,b,d)}}\right). \tag{32}$$

Where we defined

$$\begin{aligned}
\Lambda_4(a,b) &:= (\boldsymbol{C}_{aa}+1)(\boldsymbol{C}_{bb}+1) - \boldsymbol{C}_{ab}^2,\\
\Lambda_0(a,b,c,d) &:= \Lambda_4\boldsymbol{C}_{cd} - \boldsymbol{C}_{ac}\boldsymbol{C}_{ad}(\boldsymbol{C}_{bb}+1) - \boldsymbol{C}_{bc}\boldsymbol{C}_{bd}(\boldsymbol{C}_{aa}+1) + \boldsymbol{C}_{ab}(\boldsymbol{C}_{ad}\boldsymbol{C}_{bc} + \boldsymbol{C}_{bd}\boldsymbol{C}_{ac}),\\
\Lambda_1(a,b,c) &:= \Lambda_4(\boldsymbol{C}_{cc}+1) - \boldsymbol{C}_{ac}^2(\boldsymbol{C}_{bb}+1) - \boldsymbol{C}_{bc}^2(\boldsymbol{C}_{aa}+1) + 2\boldsymbol{C}_{ab}\boldsymbol{C}_{ac}\boldsymbol{C}_{bc},\\
\Lambda_2(a,b,d) &:= \Lambda_4(\boldsymbol{C}_{dd}+1) - \boldsymbol{C}_{ad}^2(\boldsymbol{C}_{bb}+1) - \boldsymbol{C}_{bd}^2(\boldsymbol{C}_{aa}+1) + 2\boldsymbol{C}_{ab}\boldsymbol{C}_{ad}\boldsymbol{C}_{bd},\\
\Lambda_3(a,c) &:= (\boldsymbol{C}_{aa}+1)(\boldsymbol{C}_{cc}+1) - \boldsymbol{C}_{ac}^2.
\end{aligned}$$

B STANDARD SEQUENTIAL TRAINING

The results presented for the training on Task 1 are identical to those presented in Lee et al. (2021; 2022) and also apply to the first phase of this LoRA-based framework. We choose to state them in this Appendix for completeness. We also report the theoretical ODEs needed to reproduce the standard approach when training on Task 2, which serve as a comparison with the theoretical results found in this work.

LIST OF ORDER PARAMETERS

We start by defining the preactivation fields of the mth teacher † unit, p^{th} teacher ‡ unit and k^{th} student unit respectively as

$$\rho_m = \frac{\boldsymbol{W}_m^\dagger \boldsymbol{\xi}}{\sqrt{N}}, \quad \nu_p = \frac{\boldsymbol{W}_p^\ddagger \boldsymbol{\xi}}{\sqrt{N}}, \quad x_k = \frac{\boldsymbol{J}_k \boldsymbol{\xi}}{\sqrt{N}}. \tag{33}$$

The set of time-dependent order parameters we recover during the first part of training are then

$$\text{Student - Student Overlap, } \boldsymbol{Q}_{kl} := \langle x_k x_l \rangle = \frac{1}{N} \boldsymbol{J}_k \boldsymbol{J}_l^T, \tag{34}$$

$$\text{Student - Teacher}^\dagger \text{ Overlap, } \boldsymbol{R}_{km} := \langle x_k \rho_m \rangle = \frac{1}{N} \boldsymbol{J}_k (\boldsymbol{W}_m^\dagger)^T, \tag{35}$$

$$\text{Student - Teacher}^\ddagger \text{Overlap, } \boldsymbol{U}_{kp} := \langle x_k \nu_p \rangle = \frac{1}{N} \boldsymbol{J}_k (\boldsymbol{W}_p^\ddagger)^T. \tag{36}$$

While the static order parameters, totally defined by the sampling procedure for the teachers given in 3 are given by

$$\text{Teacher}^\dagger \text{ - Teacher}^\dagger \text{ Overlap, } \boldsymbol{T}_{nm} := \langle \rho_n \rho_m \rangle = \frac{1}{N} \boldsymbol{W}_n^\dagger \left(\boldsymbol{W}^\dagger\right)_m^T, \tag{37}$$

$$\text{Teacher}^\ddagger \text{ - Teacher}^\ddagger \text{ Overlap, } \boldsymbol{S}_{pq} := \langle \nu_p \nu_q \rangle = \frac{1}{N} \boldsymbol{W}_p^\ddagger \left(\boldsymbol{W}^\ddagger\right)_q^T, \tag{38}$$

$$\text{Teacher}^\dagger \text{ - Teacher}^\ddagger \text{ Overlap, } \boldsymbol{V}_{mp} := \langle \rho_m \nu_p \rangle = \frac{1}{N} \boldsymbol{W}_m^\dagger \left(\boldsymbol{W}^\ddagger\right)_p^T. \tag{39}$$

GENERALIZATION ERRORS

Having defined the order parameters and preactivation fields, we are now ready to compute the generalization error on both tasks during the first part of training.

Recall that computing an average over the distribution of the input $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I}_N)$ is not needed when the functions involved depend only on the preactivations. This means we want to focus on the joint distribution of such preactivations fields

$$(x_1, \ldots, x_K, \rho_1, \ldots, \rho_M, \nu_1, \ldots, \nu_M). \tag{40}$$

In the limit $N \to \infty$, the preactivations fields are jointly gaussian and we need only to focus on the time-dependent joint covariance matrix of those preactivations, written as

$$\tilde{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{R} & \boldsymbol{U} \\ \boldsymbol{R}^T & \boldsymbol{T} & \boldsymbol{V} \\ \boldsymbol{U}^T & \boldsymbol{V}^T & \boldsymbol{S} \end{bmatrix}. \tag{41}$$

We first express the generalization errors on both tasks in term of the preactivations, giving

$$\epsilon^\dagger = \frac{1}{2} \left\langle \sum_{k,l=1}^{K} h_k^\dagger h_l^\dagger g(x_k) g(x_l) + \sum_{m,n=1}^{M} \boldsymbol{v}_m^\dagger \boldsymbol{v}_n^\dagger g(\rho_m) g(\rho_n) - 2 \sum_{k=1}^{K} \sum_{m=1}^{M} \boldsymbol{h}_k^\dagger \boldsymbol{v}_m^\dagger g(x_k) g(\rho_m) \right\rangle, \tag{42}$$

$$\epsilon^\ddagger = \frac{1}{2} \left\langle \sum_{k,l=1}^{K} h_k^\ddagger h_l^\ddagger g(x_k) g(x_l) + \sum_{p,q=1}^{M} \boldsymbol{v}_p^\ddagger \boldsymbol{v}_q^\ddagger g(\nu_p) g(\nu_q) - 2 \sum_{k=1}^{K} \sum_{p=1}^{M} \boldsymbol{h}_k^\ddagger \boldsymbol{v}_p^\ddagger g(x_k) g(\nu_p) \right\rangle. \tag{43}$$

Using the Gaussian integrals, we obtain a closed form solution for the errors

$$\epsilon^\dagger = \frac{1}{2} \sum_{k,l=1}^{K} \boldsymbol{h}_k^\dagger \boldsymbol{h}_l^\dagger I_2(k,l) + \frac{1}{2} \sum_{m,n=1}^{M} \boldsymbol{v}_m^\dagger \boldsymbol{v}_n^\dagger I_2(K+m, K+n) - \sum_{k=1}^{K} \sum_{m=1}^{M} \boldsymbol{h}_k^\dagger \boldsymbol{v}_m^\dagger I_2(k, K+m), \tag{44}$$

$$\epsilon^\ddagger = \frac{1}{2} \sum_{k,l=1}^{K} \boldsymbol{h}_k^\ddagger \boldsymbol{h}_l^\ddagger I_2(k,l) + \frac{1}{2} \sum_{p,q=1}^{M} \boldsymbol{v}_p^\ddagger \boldsymbol{v}_q^\ddagger I_2(K+M+p, K+M+q) - \sum_{k=1}^{K} \sum_{p=1}^{M} \boldsymbol{h}_k^\ddagger \boldsymbol{v}_p^\ddagger I_2(k, K+M+p). \tag{45}$$

where the variables of the various gaussian integrals represent the associaed entries of the covariance matrix (41).

GRADIENT UPDATES

In this section, we report the weight updates while training on the two tasks in the classical setting of Lee et al. (2021). In this case, the output of the student network on task $* \in \{\dagger, \ddagger\}$ is given by

$$\phi(\boldsymbol{\xi}^\mu; \boldsymbol{J}, \boldsymbol{h}^*) = \sum_{k=1}^{K} \boldsymbol{h}_k^* g\left(\frac{\boldsymbol{J}_k \boldsymbol{\xi}^\mu}{\sqrt{N}}\right). \tag{46}$$

The loss on input $\boldsymbol{\xi}^\mu$ explicitly reads

$$\ell(\boldsymbol{\xi}^\mu; \boldsymbol{J}, \boldsymbol{h}^*, \boldsymbol{W}^*, \boldsymbol{v}^*) = \frac{1}{2}\left(\sum_{m=1}^{M} \boldsymbol{v}_m^* \, g\left(\frac{\boldsymbol{W}_m^* \boldsymbol{\xi}^\mu}{\sqrt{N}}\right) - \sum_{k=1}^{K} \boldsymbol{h}_k^* \, g\left(\frac{\boldsymbol{J}_k \boldsymbol{\xi}^\mu}{\sqrt{N}}\right)\right)^2. \tag{47}$$

We define the prediction error on the μ-th example for task $*$ as

$$\Delta^{*,\mu} = \sum_{k=1}^{K} \left(\boldsymbol{h}_k^*\right)^\mu g(x_k^\mu) - \sum_{m=1}^{M} \boldsymbol{v}_m^* g_m^*, \quad g_m^\dagger := g(\rho_m); \; g_m^\ddagger := g(\nu_m). \tag{48}$$

This allows us to compute the gradient with respect to the i-th row of $\boldsymbol{J}$

$$\nabla_{\boldsymbol{J}_i} \ell(\boldsymbol{\xi}^\mu; \boldsymbol{J}, \boldsymbol{h}^*, \boldsymbol{W}^*, \boldsymbol{v}^*) = \Delta^{*,\mu} \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) \frac{(\boldsymbol{\xi}^\mu)^T}{\sqrt{N}} \tag{49}$$

from which the gradient update for $\boldsymbol{J}_i$ reads

$$\boldsymbol{J}_i^{\mu+1} = \boldsymbol{J}_i^\mu - \frac{\eta_{\boldsymbol{J}}}{\sqrt{N}} \Delta^{*,\mu} \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) (\boldsymbol{\xi}^\mu)^T. \tag{50}$$

In the same fashion, the gradient update for $\boldsymbol{h}^*$ is given by

$$\left(\boldsymbol{h}_i^*\right)^{\mu+1} = \left(\boldsymbol{h}_i^*\right)^\mu - \frac{\eta_{\boldsymbol{h}}}{N} \Delta^{*,\mu} g(x_i^\mu). \tag{51}$$

DIFFERENTIAL EQUATIONS

In the following, we will make use of the following notation:

$$\begin{aligned} \text{delay}^\dagger &= 0, \\ \text{delay}^\ddagger &= M. \end{aligned} \tag{52}$$

All integrals computed in this section are performed on the covariance matrix of standard training (41). The delays (52) are needed because the relevant preactivation fields of the teacher in the covariance matrix depend on the active task $* \in \{\dagger, \ddagger\}$ we are considering when solving the ODEs.

$\boldsymbol{Q}$: From the gradient update (50), multiplying by $(\boldsymbol{J}_k^{\mu+1})^T$ on the right and using the corresponding expressions for the preactivations:

$$\begin{aligned} \boldsymbol{J}_i^{\mu+1} (\boldsymbol{J}_k^{\mu+1})^T =& \boldsymbol{J}_i^\mu (\boldsymbol{J}_k^\mu)^T - \eta_{\boldsymbol{J}} \, \Delta^{*,\mu} \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) \, x_k^\mu - \eta_{\boldsymbol{J}} \, \Delta^{*,\mu} \left(\boldsymbol{h}_k^*\right)^\mu g'(x_k^\mu) \, x_i^\mu \\ &+ \eta_{\boldsymbol{J}}^2 \left(\Delta^{*,\mu}\right)^2 \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) \left(\boldsymbol{h}_k^*\right)^\mu g'(x_k^\mu) \, \frac{||\boldsymbol{\xi}^\mu||^2}{N}. \end{aligned}$$

Rearranging terms and substituting the order parameters, we obtain

$$\begin{aligned} \frac{\boldsymbol{Q}_{ik}^{\mu+1} - \boldsymbol{Q}_{ik}^\mu}{1/N} =& -\eta_{\boldsymbol{J}} \, \Delta^{*,\mu} \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) \, x_k^\mu - \eta_{\boldsymbol{J}} \, \Delta^{*,\mu} \left(\boldsymbol{h}_k^*\right)^\mu g'(x_k^\mu) \, x_i^\mu \\ &+ \eta_{\boldsymbol{J}}^2 (\Delta^{*,\mu})^2 \left(\boldsymbol{h}_i^*\right)^\mu g'(x_i^\mu) \left(\boldsymbol{h}_k^*\right)^\mu g'(x_k^\mu) \, \frac{||\boldsymbol{\xi}^\mu||^2}{N}. \end{aligned}$$

Defining $\tau := \mu/N$ and taking the thermodynamic limit $N \to \infty$, the discrete difference equation converges to the continuous-time differential equation:

$$\frac{\mathrm{d}\boldsymbol{Q}_{ik}}{\mathrm{d}\tau} = -\eta_{\boldsymbol{J}}\, \boldsymbol{h}_i^* \,\langle g'(x_i)\, x_k\, \Delta^* \rangle - \eta_{\boldsymbol{J}}\, \boldsymbol{h}_k^* \,\langle g'(x_k)\, x_i\, \Delta^* \rangle + \eta_{\boldsymbol{J}}^2\, \boldsymbol{h}_i^*\, \boldsymbol{h}_k^* \,\langle g'(x_i) g'(x_k) (\Delta^*)^2 \rangle$$

that we can rewrite in terms of the integrals found in Appendix A after explicitly substituting Δ^* and $(\Delta^*)^2$:

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{Q}_{ik}}{\mathrm{d}\tau} =& \eta_{\boldsymbol{J}}\, \boldsymbol{h}_i^* \left[\sum_{m=1}^{M} \boldsymbol{v}_m^*\, I_3(i,k,K+\mathrm{delay}^*+m) - \sum_{j=1}^{K} \boldsymbol{h}_j^*\, I_3(i,k,j) \right] \\
&+ \eta_{\boldsymbol{J}}\, \boldsymbol{h}_k^* \left[\sum_{m=1}^{M} \boldsymbol{v}_m^*\, I_3(k,i,K+\mathrm{delay}^*+m) - \sum_{j=1}^{K} \boldsymbol{h}_j^*\, I_3(k,i,j) \right] \\
&+ \eta_{\boldsymbol{J}}^2\, \boldsymbol{h}_i^*\, \boldsymbol{h}_k^* \left[\sum_{j,l=1}^{K} \boldsymbol{h}_j^*\, \boldsymbol{h}_l^*\, I_4(i,k,j,l) \right. \\
&+ \sum_{m,n=1}^{M} \boldsymbol{v}_m^*\, \boldsymbol{v}_n^*\, I_4(i,k,K+\mathrm{delay}^*+m,K+\mathrm{delay}^*+n) \\
&\left. - 2 \sum_{j=1}^{K} \sum_{m=1}^{M} \boldsymbol{h}_j^*\, \boldsymbol{v}_m^*\; I_4(i,k,j,K+\mathrm{delay}^*+m) \right].
\end{aligned} \tag{53}$$

$\boldsymbol{R}$**:** From the gradient update 50, multiplying by $\boldsymbol{W}_n^{\dagger\,T}$ on the right and using the corresponding expressions for the relevant preactivations:

$$\boldsymbol{J}_i^{\mu+1} \boldsymbol{W}_n^{\dagger\,T} = \boldsymbol{J}_i^{\mu} \boldsymbol{W}_n^{\dagger\,T} - \eta_{\boldsymbol{J}}\, \Delta^{*\mu}\, \boldsymbol{h}_i^{*\mu}\, g'(x_i^\mu)\, \rho_n^\mu$$

Rearranging terms and substituting the order parameter, we obtain:

$$\frac{\boldsymbol{R}_{in}^{\mu+1} - \boldsymbol{R}_{in}^{\mu}}{1/N} = -\eta_{\boldsymbol{J}}\, \Delta^{*,\mu}\, \left(\boldsymbol{h}_i^*\right)^{\mu}\, g'(x_i^\mu)\, \rho_n^\mu.$$

Performing the thermodynamic limit $N \to \infty$ we obtain the differential equation:

$$\frac{\mathrm{d}\boldsymbol{R}_{in}}{\mathrm{d}\tau} = \eta_{\boldsymbol{J}}\, \boldsymbol{h}_i^* \left[\sum_{m=1}^{M} \boldsymbol{v}_m^*\, I_3(i,K+n,K+\mathrm{delay}^*+m) - \sum_{j=1}^{K} \boldsymbol{h}_j^*\, I_3(i,K+n,j) \right], \tag{54}$$

$\boldsymbol{h}^*$**:** From the gradient update 51 we can write

$$\frac{\boldsymbol{h}_i^{*\mu+1} - \boldsymbol{h}_i^{*\mu}}{1/N} = -\eta_{\boldsymbol{h}} \Delta^{*,\mu} g(x_i^\mu)$$

and take the thermodynamic limit to obtain

$$\frac{\mathrm{d}\boldsymbol{h}_i^*}{\mathrm{d}\tau} = \eta_{\boldsymbol{h}} \left[\sum_{m=1}^{M} \boldsymbol{v}_m^*\, I_2(K+\mathrm{delay}^*+m,i) - \sum_{j=1}^{K} \boldsymbol{h}_j^*\, I_2(j,i) \right]. \tag{55}$$

$\boldsymbol{U}$**:** From the gradient update 50, multiplying by $\boldsymbol{W}_p^{\ddagger\,T}$ on the right and using the corresponding expressions for the relevant preactivations:

$$\boldsymbol{J}_i^{\mu+1} \boldsymbol{W}_p^{\ddagger\,T} = \boldsymbol{J}_i^{\mu} \boldsymbol{W}_p^{\ddagger\,T} - \eta_{\boldsymbol{J}}\, \Delta^{*,\mu}\, \left(\boldsymbol{h}_i^*\right)^{\mu}\, g'(x_i^\mu)\, \nu_p^\mu$$

Rearranging terms and substituting the order parameter, we obtain:

$$\frac{\boldsymbol{U}_{ip}^{\mu+1} - \boldsymbol{U}_{ip}^{\mu}}{1/N} = -\eta_{\boldsymbol{J}}\, \Delta^{*,\mu}\, \left(\boldsymbol{h}_i^*\right)^{\mu}\, g'(x_i^\mu)\, \nu_p^\mu.$$

Performing the thermodynamic limit $N \to \infty$ we obtain the differential equation:

$$\frac{\mathrm{d}\boldsymbol{U}_{ip}}{\mathrm{d}\tau} = \eta_{\boldsymbol{J}}\, \boldsymbol{h}_i^* \left[\sum_{m=1}^{M} \boldsymbol{v}_m^*\, I_3(i, K+M+p, K+\mathrm{delay}^*+m) - \sum_{j=1}^{K} \boldsymbol{h}_j^*\, I_3(i, K+M+p, j) \right]. \tag{56}$$

C Low-rank Adaptation on Task 2

This section presents the new differential equations arising from Low-Rank Adaptation in the online learning paradigm. The equations given in this appendix must be used when training on Task 2 only. To retrieve $\boldsymbol{Q}_s, \boldsymbol{R}_s, \boldsymbol{U}_s, \boldsymbol{h}^\dagger$, the ODEs given in Appendix B must first be integrated (by setting $* = \dagger$). In the following, we use the shortcut $\beta := \gamma/\sqrt{L}$.

List of order parameters

By defining the preactivation fields of the mth teacher $\dagger$ unit, p^{th} teacher $\ddagger$ unit, k^{th} student unit, and i^{th} direction of the LoRA update of the student unit respectively as

$$\rho_m = \frac{\boldsymbol{W}_m^\dagger \boldsymbol{\xi}}{\sqrt{N}}, \quad \nu_p = \frac{\boldsymbol{W}_p^\ddagger \boldsymbol{\xi}}{\sqrt{N}}, \quad x_k = \frac{(\boldsymbol{J}_s)_k \boldsymbol{\xi}}{\sqrt{N}}, \quad z_i = \frac{\boldsymbol{A}_i \boldsymbol{\xi}}{\sqrt{N}}, \quad \tilde{x}_k = x_k + \beta \sum_{l=1}^{L} \boldsymbol{B}_{kl} z_l. \tag{57}$$

Note that after the task switch, x_k is a constant depending on the frozen weight $(\boldsymbol{J}_s)_k$ only.
Hence, the full set of time-dependent order parameters recovered by the theory in the second part of training are:

$$\begin{aligned}
&\text{Down-projection–Down-projection Overlap, } : \boldsymbol{\Phi}_{ij} := \langle z_i z_j \rangle = \frac{1}{N} \boldsymbol{A}_i \boldsymbol{A}_j^T \\
&\text{Teacher}^\dagger\text{–Down-projection Overlap, } : \boldsymbol{\Lambda}_{mi} := \langle \rho_m z_i \rangle = \frac{1}{N} \boldsymbol{W}_m^\dagger \boldsymbol{A}_i^T \\
&\text{Teacher}^\ddagger\text{–Down-projection Overlap, } : \boldsymbol{\Gamma}_{pi} := \langle \nu_p z_i \rangle = \frac{1}{N} \boldsymbol{W}_p^\ddagger \boldsymbol{A}_i^T \\
&\text{Student–Down-projection Overlap, } : \boldsymbol{\Xi}_{ki} := \langle x_k z_i \rangle = \frac{1}{N} (\boldsymbol{J}_s)_k \boldsymbol{A}_i^T \\
&\text{Up-projection Matrix, } \boldsymbol{B}.
\end{aligned}$$

while the other order parameters

$$\begin{aligned}
&\boldsymbol{Q}_{kl} := \langle x_k x_l \rangle, \\
&\boldsymbol{R}_{km} := \langle x_k \rho_m \rangle, \\
&\boldsymbol{U}_{kp} := \langle x_k \nu_p \rangle, \\
&\boldsymbol{T}_{mn} := \langle \rho_m \rho_n \rangle, \\
&\boldsymbol{S}_{pq} := \langle \nu_p \nu_q \rangle, \\
&\boldsymbol{V}_{mp} := \langle \rho_m \nu_p \rangle
\end{aligned}$$

are frozen in this specific part of training, their evolution being tracked in the first part of training using the closed form formulae given in Lee et al. (2021) and mentioned in B.

Generalization errors

We start by computing the generalization error on both tasks during the second part of training.

Recall that computing an average over the distribution of the input $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I}_N)$ is not needed when the functions involved depend only on the preactivations. This means we want to focus on the joint distribution of such preactivations. Although $\tilde{x}_i$ is just a linear transformation of $(x_i, z_1, \cdots, z_L)$, it is recommended to consider the higher dimensional gaussian distribution of the random vector

$$(x_1, \ldots, x_K, \tilde{x}_1, \ldots, \tilde{x}_K, \nu_1, \ldots, \nu_M, z_1, \ldots, z_L, \rho_1, \ldots, \rho_M). \tag{58}$$

Starting from the definition

$$\begin{aligned}
\langle x_k x_l \rangle &= \boldsymbol{Q}_{kl}, \\
\langle x_k \nu_p \rangle &= \boldsymbol{U}_{kp}, \\
\langle x_k \rho_m \rangle &= \boldsymbol{R}_{km} \\
\langle \nu_p \nu_q \rangle &= \boldsymbol{S}_{pq}, \\
\langle \rho_m \rho_n \rangle &= \boldsymbol{T}_{mn} \\
\langle \rho_m \nu_p \rangle &= \boldsymbol{V}_{mp} \\
\langle z_i z_j \rangle &= \boldsymbol{\Phi}_{ij}, \\
\langle \nu_p z_i \rangle &= \boldsymbol{\Gamma}_{pi}, \\
\langle \rho_m z_i \rangle &= \boldsymbol{\Lambda}_{mi} \\
\langle x_k z_i \rangle &= \boldsymbol{\Xi}_{ki},
\end{aligned}$$

and computing the remaining interactions

$$\begin{aligned}
\langle \tilde{x}_k z_i \rangle &= \frac{(\boldsymbol{J}_k + \beta\, \boldsymbol{B}_k \boldsymbol{A}) \boldsymbol{A}_i^T}{N} = \frac{\boldsymbol{J}_k \boldsymbol{A}_i^T}{N} + \beta\, \boldsymbol{B}_k \frac{\boldsymbol{A}\boldsymbol{A}_i^T}{N} = \boldsymbol{\Xi}_{ki} + \beta\, \boldsymbol{B}_k\, \boldsymbol{\Phi}_i^T, \\
\langle \tilde{x}_k \nu_p \rangle &= \frac{(\boldsymbol{J}_k + \beta\, \boldsymbol{B}_k \boldsymbol{A}) \boldsymbol{W}_p^{\ddagger T}}{N} = \frac{\boldsymbol{J}_k \boldsymbol{W}_p^{\ddagger T}}{N} + \beta\, \boldsymbol{B}_k \frac{\boldsymbol{A}\boldsymbol{W}_p^{\ddagger T}}{N} = \boldsymbol{U}_{kp} + \beta \boldsymbol{B}_k \boldsymbol{\Gamma}_p^T, \\
\langle \tilde{x}_k \rho_m \rangle &= \frac{(\boldsymbol{J}_k + \beta\, \boldsymbol{B}_k \boldsymbol{A}) \boldsymbol{W}_m^{\dagger\, T}}{N} = \frac{\boldsymbol{J}_k \boldsymbol{W}_m^{\dagger\, T}}{N} + \beta\, \boldsymbol{B}_k \frac{\boldsymbol{A}\boldsymbol{W}_m^{\dagger\, T}}{N} = \boldsymbol{R}_{km} + \beta \boldsymbol{B}_k \boldsymbol{\Lambda}_m^T, \\
\langle x_k \tilde{x}_l \rangle &= \frac{\boldsymbol{J}_k (\boldsymbol{J}_l + \beta \boldsymbol{B}_l \boldsymbol{A})^T}{N} = \frac{\boldsymbol{J}_k \boldsymbol{J}_l^T}{N} + \beta \frac{\boldsymbol{J}_k \boldsymbol{A}^T \boldsymbol{B}_l^T}{N} = \boldsymbol{Q}_{kl} + \beta\, \boldsymbol{\Xi}_{k:}\, \boldsymbol{B}_l^T, \\
\langle \tilde{x}_k \tilde{x}_l \rangle &= \frac{\tilde{\boldsymbol{J}}_k \tilde{\boldsymbol{J}}_l^T}{N} = \frac{1}{N} \left(\boldsymbol{J}_k + \beta\, \boldsymbol{B}_k \boldsymbol{A}\right) \left(\boldsymbol{J}_l^T + \beta\, \boldsymbol{A}^T \boldsymbol{B}_l^T\right) \\
&= \frac{\boldsymbol{J}_k \boldsymbol{J}_l^T}{N} + \beta\, \frac{\boldsymbol{J}_k \boldsymbol{A}^T}{N} \boldsymbol{B}_l^T + \beta\, \boldsymbol{B}_k\, \frac{\boldsymbol{A}\boldsymbol{J}_l^T}{N} + \beta^2\, \boldsymbol{B}_k\, \frac{\boldsymbol{A}\boldsymbol{A}^T}{N} \boldsymbol{B}_l^T \\
&= \boldsymbol{Q}_{kl} + \beta\, \boldsymbol{\Xi}_k \boldsymbol{B}_l^T + \beta\, \boldsymbol{B}_k \boldsymbol{\Xi}_l^T + \beta^2\, \boldsymbol{B}_k \boldsymbol{\Phi} \boldsymbol{B}_l^T,
\end{aligned}$$

allows us to determine the covariance matrix of the $(2K + 2M + L)$-dimensional gaussian vector (58)

$$\boldsymbol{C} = \begin{bmatrix}
\boldsymbol{Q} & \boldsymbol{Q} + \beta\, \boldsymbol{\Xi}\, \boldsymbol{B}^T & \boldsymbol{U} & \boldsymbol{\Xi} & \boldsymbol{R} \\
\boldsymbol{Q}^T + \beta\, \boldsymbol{B}\boldsymbol{\Xi}^T & \boldsymbol{Q} + \beta(\boldsymbol{\Xi}\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{\Xi}^T) + \beta^2 \boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{B}^T & \boldsymbol{U} + \beta\, \boldsymbol{B}\boldsymbol{\Gamma}^T & \boldsymbol{\Xi} + \beta\, \boldsymbol{B}\boldsymbol{\Phi}^T & \boldsymbol{R} + \beta \boldsymbol{B}\boldsymbol{\Lambda}^T \\
\boldsymbol{U}^T & \boldsymbol{U}^T + \beta\, \boldsymbol{\Gamma}\boldsymbol{B}^T & \boldsymbol{S} & \boldsymbol{\Gamma} & \boldsymbol{V}^T \\
\boldsymbol{\Xi}^T & \boldsymbol{\Xi}^T + \beta\, \boldsymbol{\Phi}\boldsymbol{B}^T & \boldsymbol{\Gamma}^T & \boldsymbol{\Phi} & \boldsymbol{\Lambda}^T \\
\boldsymbol{R}^T & \boldsymbol{R}^T + \beta \boldsymbol{\Lambda}\boldsymbol{B}^T & \boldsymbol{V} & \boldsymbol{\Lambda} & \boldsymbol{T}
\end{bmatrix}. \tag{59}$$

We can write the generalization error on the second task in terms of the preactivations:

$$\begin{aligned}
\epsilon^{\ddagger}\left(\boldsymbol{h}^{\ddagger}, \boldsymbol{J}, \boldsymbol{v}^{\ddagger}, \boldsymbol{W}^{\ddagger}\right) &= \frac{1}{2} \left\langle \left(\sum_{p=1}^{M} \boldsymbol{v}_p^{\ddagger}\, g\!\left(\frac{\boldsymbol{W}_p^{\ddagger} \xi}{\sqrt{N}}\right) - \sum_{k=1}^{K} \boldsymbol{h}_k^{\ddagger}\, g\left(\frac{\boldsymbol{J}_k \xi + \beta\, \boldsymbol{B}_k \boldsymbol{A} \xi}{\sqrt{N}}\right) \right)^2 \right\rangle \\
&= \frac{1}{2} \left\langle \left(\sum_{p=1}^{M} \boldsymbol{v}_p^{\ddagger}\, g\left(\nu_p\right) - \sum_{k=1}^{K} \boldsymbol{h}_k^{\ddagger}\, g\left(\tilde{x}_k\right) \right)^2 \right\rangle \\
&= \frac{1}{2} \sum_{p,q=1}^{M} \boldsymbol{v}_p^{\ddagger} \boldsymbol{v}_q^{\ddagger} \big\langle g(\nu_p) g(\nu_q) \big\rangle \\
&\quad - \sum_{p=1}^{M} \sum_{k=1}^{K} \boldsymbol{v}_p^{\ddagger} \boldsymbol{h}_k^{\ddagger} \big\langle g(\nu_p) g(\tilde{x}_k) \big\rangle \\
&\quad + \frac{1}{2} \sum_{k,l=1}^{K} \boldsymbol{h}_k^{\ddagger} \boldsymbol{h}_l^{\ddagger} \big\langle g(\tilde{x}_k) g(\tilde{x}_l) \big\rangle
\end{aligned}$$

Expressing this quantity as a function of the gaussian integrals given in Appendix A allows us to cancel the explicit dependency on the first-layers and to close the equation

$$\begin{aligned}\epsilon^{\ddagger}\left(\boldsymbol{h}^{\ddagger},\boldsymbol{v}^{\ddagger}\right) = &\frac{1}{2}\sum_{p,q=1}^{M}\boldsymbol{v}_p^{\ddagger}\boldsymbol{v}_q^{\ddagger}I_2(2K+p,2K+q)\\ &-\sum_{p=1}^{M}\sum_{k=1}^{K}\boldsymbol{v}_p^{\ddagger}\boldsymbol{h}_k^{\ddagger}I_2(2K+p,K+k)\\ &+\frac{1}{2}\sum_{k,l=1}^{K}\boldsymbol{h}_k^{\ddagger}\boldsymbol{h}_l^{\ddagger}I_2(K+k,K+l).\end{aligned} \tag{60}$$

In the same fashion, the generalization error on the first task results in

$$\begin{aligned}\epsilon^{\dagger}\left(\boldsymbol{h}^{\dagger},\boldsymbol{v}^{\dagger}\right) = &\frac{1}{2}\sum_{p,q=1}^{M}\boldsymbol{v}_p^{\dagger}\boldsymbol{v}_q^{\dagger}I_2(2K+M+L+p,2K+M+L+q)\\ &-\sum_{p=1}^{M}\sum_{k=1}^{K}\boldsymbol{v}_p^{\dagger}\boldsymbol{h}_k^{\dagger}I_2(2K+M+L+p,K+k)\\ &+\frac{1}{2}\sum_{k,l=1}^{K}\boldsymbol{h}_k^{\dagger}\boldsymbol{h}_l^{\dagger}I_2(K+k,K+l).\end{aligned} \tag{61}$$

GRADIENT UPDATES

We start by expliciting the gradient updates of the LoRA adapter.

Let $\boldsymbol{\xi}^{\mu}$ be the input vector of the network and $\beta := \gamma/\sqrt{L}$ the LoRA scaling factor. The output of the student network on the second task is given by

$$\phi(\boldsymbol{\xi}^{\mu};\boldsymbol{J}_s,\boldsymbol{B},\boldsymbol{A},\boldsymbol{h}^{\ddagger}) = \sum_{k=1}^{K}\boldsymbol{h}_k^{\ddagger\,\mu}\,g\left(\frac{(\boldsymbol{J}_s)_k\boldsymbol{\xi}^{\mu}+\beta\boldsymbol{B}_k^{\mu}\boldsymbol{A}^{\mu}\boldsymbol{\xi}^{\mu}}{\sqrt{N}}\right). \tag{62}$$

Where $(\boldsymbol{J}_s)_k$ and $\boldsymbol{B}_k$ are the k-th rows of $\boldsymbol{J}_s$ and $\boldsymbol{B}$, respectively. We also define $\tilde{\boldsymbol{J}}_k = (\boldsymbol{J}_s)_k + \beta\boldsymbol{B}_k\mathbf{A}$. The loss on input $\boldsymbol{\xi}^{\mu}$ can be rewritten as

$$\ell(\boldsymbol{\xi}^{\mu};\boldsymbol{h}^{\ddagger},\boldsymbol{J}_s,\boldsymbol{B},\boldsymbol{A},\boldsymbol{v},\boldsymbol{W}^{\ddagger}) = \frac{1}{2}\left(\sum_{m=1}^{M}\boldsymbol{v}_m^{\ddagger}\,g\left(\frac{\boldsymbol{W}_m^{\ddagger}\xi^{\mu}}{\sqrt{N}}\right)-\sum_{k=1}^{K}\boldsymbol{h}_k^{\ddagger\,\mu}\,g\left(\frac{\boldsymbol{J}_k\boldsymbol{\xi}^{\mu}+\beta\boldsymbol{B}_k^{\mu}\boldsymbol{A}^{\mu}\boldsymbol{\xi}^{\mu}}{\sqrt{N}}\right)\right)^2. \tag{63}$$

Using the preactivations, we can compute the partial derivative of 63 with respect to $\boldsymbol{B}_{ki}$:

$$\begin{aligned}\partial_{\boldsymbol{B}_{ki}}\ell(\boldsymbol{\xi}^{\mu};\boldsymbol{h}^{\ddagger},\boldsymbol{J}_s,\boldsymbol{B},\boldsymbol{A},\boldsymbol{v}^{\ddagger},\boldsymbol{W}^{\ddagger}) &= (\Delta^{\ddagger})^{\mu}\partial_{\boldsymbol{B}_{ki}}\left(\sum_{l=1}^{K}\boldsymbol{h}_l^{\ddagger\,\mu}\,g\left(x_l^{\mu}+\beta\sum_{j=1}^{L}\boldsymbol{B}_{lj}^{\mu}z_j^{\mu}\right)\right)\\ &= \beta(\Delta^{\ddagger})^{\mu}\boldsymbol{h}_k^{\ddagger\,\mu}g'\left(\tilde{x}_k^{\mu}\right)z_i^{\mu}.\end{aligned} \tag{64}$$

The gradient update for $\boldsymbol{B}$ is therefore

$$\boldsymbol{B}_{ki}^{\mu+1} = \boldsymbol{B}_{ki}^{\mu} - \beta\frac{\eta_{\boldsymbol{B}}}{N}(\Delta^{\ddagger})^{\mu}\boldsymbol{h}_k^{\ddagger\,\mu}g'\left(\tilde{x}_k^{\mu}\right)z_i^{\mu} \tag{65}$$

where an extra prefactor $1/N$ has been added in the gradient update rule of $\boldsymbol{B}$ in the same fashion as the update rule for the readout weights (51, 69), such that the scaling in $1/N$ allows for a well-defined, non-trivial thermodynamic limit.

At the same time, taking the partial derivative of the loss (63) with respect to $\boldsymbol{A}_{in}$ results in

$$\partial_{\boldsymbol{A}_{in}}\ell = (\Delta^{\ddagger})^{\mu}\sum_{k=1}^{K}\boldsymbol{h}_k^{\ddagger}\,g'\left(\tilde{x}_k\right)\frac{\beta}{\sqrt{N}}\boldsymbol{B}_{ki}\boldsymbol{\xi}_n^{\mu}. \tag{66}$$

In matrix form, if $\boldsymbol{A}_i$ is the i-th row of $\mathbf{A}$, we end up with

$$\boldsymbol{A}_i^{\mu+1} = \boldsymbol{A}_i^{\mu} - \beta \frac{\eta_{\boldsymbol{A}}}{\sqrt{N}} (\Delta^{\ddagger})^{\mu} \left(\sum_{k=1}^{K} \boldsymbol{h}_k^{\ddagger\,\mu} g'\left(\tilde{x}_k^{\mu}\right) \boldsymbol{B}_{ki}^{\mu} \right) (\boldsymbol{\xi}^{\mu})^T. \tag{67}$$

DIFFERENTIAL EQUATIONS

We are now ready to recover the various differential equations tracking the evolutions of the various order parameters. In the following, when writing $\boldsymbol{v}$, $\boldsymbol{h}$, Δ or $\boldsymbol{W}_p$, we implicitly refer to the quantities associated with the second task, namely $\boldsymbol{v}^{\ddagger}$, $\boldsymbol{h}^{\ddagger}$, $\Delta^{\ddagger}$ and $\boldsymbol{W}_p^{\ddagger}$, respectively.

B: We can rewrite the update rule of $\boldsymbol{B}$ (65) as

$$\frac{\boldsymbol{B}_{ki}^{\mu+1} - \boldsymbol{B}_{ki}^{\mu}}{1/N} = -\eta_{\boldsymbol{B}}\, \beta\, \Delta^{\mu} \boldsymbol{h}_k^{\mu} g'\left(\tilde{x}_k^{\mu}\right) z_i^{\mu}.$$

Defining $\tau := \mu/N$ and taking the thermodynamic limit $N, \mu \to \infty$ with τ fixed, the discrete dynamics converge to a continuous-time evolution. The corresponding differential equation for $\boldsymbol{B}$ is

$$\frac{\mathrm{d}\boldsymbol{B}_{ki}}{\mathrm{d}\tau} = -\eta_{\boldsymbol{B}}\, \beta\, \boldsymbol{h}_k \left\langle g'\left(\tilde{x}_k\right)\, \Delta\, z_i \right\rangle.$$

In this limit, an average over the preactivation can be explicitly taken, resulting in a deterministic time evolution. As for the generalization error, we can rewrite the update as a function of the gaussian integrals given in Appendix A:

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{B}_{ki}}{\mathrm{d}\tau} &= \eta_{\boldsymbol{B}}\, \beta\, \boldsymbol{h}_k \left[\sum_{p=1}^{M} \boldsymbol{v}_p \left\langle g'\left(\tilde{x}_k\right)\, z_i\, g\left(\nu_p\right) \right\rangle - \sum_{l=1}^{K} \boldsymbol{h}_l \left\langle g'\left(\tilde{x}_k\right)\, z_i\, g\left(\tilde{x}_l\right) \right\rangle \right] \\
&= \eta_{\boldsymbol{B}}\, \beta\, \boldsymbol{h}_k \left[\sum_{p=1}^{M} \boldsymbol{v}_p I_3(K+k, 2K+M+i, 2K+p) \right. \\
&\qquad \left. - \sum_{l=1}^{K} \boldsymbol{h}_l I_3(K+k, 2K+M+i, K+l) \right].
\end{aligned} \tag{68}$$

$\boldsymbol{h}$: From the loss on the second task (63), the gradient update for $\boldsymbol{h}$ states

$$\boldsymbol{h}_k^{\mu+1} = \boldsymbol{h}_k^{\mu} - \frac{\eta_{\boldsymbol{h}}}{N} \Delta^{\mu} g(\tilde{x}_k^{\mu}). \tag{69}$$

Rearranging the terms of the gradient update and taking the thermodynamic limit, the differential equation for the readout weights reads

$$\frac{\mathrm{d}\boldsymbol{h}_k}{\mathrm{d}\tau} = \eta_{\boldsymbol{h}} \left[\sum_{p=1}^{M} \boldsymbol{v}_p\, I_2(2K+p, K+k) - \sum_{l=1}^{K} \boldsymbol{h}_l\, I_2(K+l, K+k) \right]. \tag{70}$$

$\boldsymbol{\Phi}$: Starting from the update rule of $\boldsymbol{A}$ (67) and multiplying by $(\boldsymbol{A}_j^{\mu+1})^T$ on the right, we get

$$\begin{aligned}
\boldsymbol{A}_i^{\mu+1} (\boldsymbol{A}_j^{\mu+1})^T &= \boldsymbol{A}_i^{\mu} (\boldsymbol{A}_j^{\mu})^T - \eta_{\boldsymbol{A}}\, \beta\, \Delta^{\mu} \left(\sum_{k=1}^{K} \boldsymbol{h}_k^{\mu} g'\left(\tilde{x}_k^{\mu}\right) \boldsymbol{B}_{ki}^{\mu} \right) \frac{\boldsymbol{A}_j^{\mu} \boldsymbol{\xi}^{\mu}}{\sqrt{N}} \\
&\quad - \eta_{\boldsymbol{A}}\, \beta\, \Delta^{\mu} \left(\sum_{k=1}^{K} \boldsymbol{h}_k^{\mu} g'\left(\tilde{x}_k^{\mu}\right) \boldsymbol{B}_{kj}^{\mu} \right) \frac{\boldsymbol{A}_i^{\mu} \boldsymbol{\xi}^{\mu}}{\sqrt{N}} \\
&\quad + \eta_{\boldsymbol{A}}^2\, \beta^2\, (\Delta^{\mu})^2 \left(\sum_{k=1}^{K} \boldsymbol{h}_k^{\mu} g'\left(\tilde{x}_k^{\mu}\right) \boldsymbol{B}_{ki}^{\mu} \right) \left(\sum_{l=1}^{K} \boldsymbol{h}_l^{\mu} g'\left(\tilde{x}_l^{\mu}\right) \boldsymbol{B}_{lj}^{\mu} \right) \frac{\|\boldsymbol{\xi}^{\mu}\|^2}{N}.
\end{aligned}$$

Rewriting this quantity as a function of the order parameters and taking the thermodynamic limit results in

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\Phi}_{ij}}{\mathrm{d}\tau} = &-\eta_{\boldsymbol{A}}\,\beta\,\left(\sum_{k=1}^{K}\boldsymbol{h}_k\boldsymbol{B}_{ki}\big\langle g'\left(\tilde{x}_k\right)\,z_j\,\Delta\big\rangle\right)\\ &-\eta_{\boldsymbol{A}}\,\beta\,\left(\sum_{k=1}^{K}\boldsymbol{h}_k\boldsymbol{B}_{kj}\big\langle g'\left(\tilde{x}_k\right)\,z_i\,\Delta\big\rangle\right)\\ &+\eta_{\boldsymbol{A}}^2\,\beta^2\,\left(\sum_{k,l=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{B}_{ki}\boldsymbol{B}_{lj}\big\langle g'\left(\tilde{x}_k\right)g'\left(\tilde{x}_l\right)\Delta^2\big\rangle\right).\end{aligned}$$

Finally, expanding the definition of Δ and Δ^2 and writing the quantities in terms of I_3 and I_4, we obtain a closed form solution for the differential equation

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\Phi}_{ij}}{\mathrm{d}\tau} = &\,\eta_{\boldsymbol{A}}\beta\left[\sum_{k=1}^{K}\sum_{p=1}^{M}\boldsymbol{h}_k\boldsymbol{v}_p\boldsymbol{B}_{ki}I_3(K+k,2K+M+j,2K+p)\right.\\ &\left.-\sum_{k,l=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{B}_{ki}I_3(K+k,2K+M+j,K+l)\right]\\ &+\eta_{\boldsymbol{A}}\beta\left[\sum_{k=1}^{K}\sum_{p=1}^{M}\boldsymbol{h}_k\boldsymbol{v}_p\boldsymbol{B}_{kj}I_3(K+k,2K+M+i,2K+p)\right.\\ &\left.-\sum_{k,l=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{B}_{kj}I_3(K+k,2K+M+i,K+l)\right]\\ &+\eta_{\boldsymbol{A}}^2\beta^2\left[\sum_{k,l=1}^{K}\sum_{k',l'=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{h}_{k'}\boldsymbol{h}_{l'}\boldsymbol{B}_{ki}\boldsymbol{B}_{lj}I_4(K+k,K+l,K+k',K+l')\right.\\ &-2\sum_{k,l=1}^{K}\sum_{k'=1}^{K}\sum_{p=1}^{M}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{h}_{k'}\boldsymbol{v}_p\boldsymbol{B}_{ki}\boldsymbol{B}_{lj}I_4(K+k,K+l,K+k',2K+p)\\ &\left.+\sum_{k,l=1}^{K}\sum_{p,q=1}^{M}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{v}_p\boldsymbol{v}_q\boldsymbol{B}_{ki}\boldsymbol{B}_{lj}I_4(K+k,K+l,2K+p,2K+q)\right].\end{aligned}\tag{71}$$

$\boldsymbol{\Xi}$: Starting from the update rule of $\boldsymbol{A}^T$ (67),multiplying by $\boldsymbol{J}_k^{\mu+1}$ on the left and substituting the definition of $\boldsymbol{\Xi}$, we get

$$\frac{\boldsymbol{\Xi}_{ki}^{\mu+1}-\boldsymbol{\Xi}_{ki}^{\mu}}{1/N} = -\eta_{\boldsymbol{A}}\beta\Delta^{\mu}\left(\sum_{l=1}^{K}\boldsymbol{h}_l^{\mu}\,g'(\tilde{x}_l^{\mu})\boldsymbol{B}_{li}^{\mu}\,x_k\right)$$

which can be rewritten in the thermodynamic limit as:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\Xi}_{ki}}{\mathrm{d}\tau} = \eta_{\boldsymbol{A}}\,\beta\,&\left[\sum_{l=1}^{K}\sum_{p=1}^{M}\boldsymbol{h}_l\boldsymbol{v}_p\boldsymbol{B}_{li}I_3(K+l,k,2K+p)\right.\\ &\left.-\sum_{l,k'=1}^{K}\boldsymbol{h}_l\boldsymbol{h}_{k'}\boldsymbol{B}_{li}I_3(K+l,k,K+k')\right].\end{aligned}\tag{72}$$

$\boldsymbol{\Gamma}$: Starting from the update rule of $\boldsymbol{A}^T$ (67),multiplying by $\boldsymbol{W}_p^{\ddagger}$ on the left and substituting the definition of $\boldsymbol{\Gamma}$, we get

$$\frac{\boldsymbol{\Gamma}_{pi}^{\mu+1}-\boldsymbol{\Gamma}_{pi}^{\mu}}{1/N} = -\eta_{\boldsymbol{A}}\beta\Delta^{\mu}\left(\sum_{k=1}^{K}\boldsymbol{h}_k^{\mu}\,g'(\tilde{x}_k^{\mu})\boldsymbol{B}_{ki}^{\mu}\,\nu_p\right)$$

which, in the thermodynamic limit, becomes

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_{pi}}{\mathrm{d}\tau} = \eta_{\boldsymbol{A}}\,\beta\,\left[\sum_{k=1}^{K}\sum_{q=1}^{M}\boldsymbol{h}_k\boldsymbol{v}_q\boldsymbol{B}_{ki}I_3(K+k,2K+p,2K+q) - \sum_{k,l=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{B}_{ki}I_3(K+k,2K+p,K+l)\right]. \tag{73}$$

$\boldsymbol{\Lambda}$: Starting from the update rule of $\boldsymbol{A}^T$ (67), multiplying by $\boldsymbol{W}_p^\dagger$ on the left and substituting the definition of $\boldsymbol{\Lambda}$, we get

$$\frac{\boldsymbol{\Lambda}_{mi}^{\mu+1}-\boldsymbol{\Lambda}_{mi}^{\mu}}{1/N} = -\eta_{\boldsymbol{A}}\beta\Delta^\mu\left(\sum_{k=1}^{K}\boldsymbol{h}_k^\mu\,g'(\tilde{x}_k^\mu)\boldsymbol{B}_{ki}^\mu\,\rho_m\right)$$

which becomes in the thermodynamic limit:

$$\frac{\mathrm{d}\boldsymbol{\Lambda}_{mi}}{\mathrm{d}\tau} = \eta_{\boldsymbol{A}}\,\beta\,\left[\sum_{k=1}^{K}\sum_{p=1}^{M}\boldsymbol{h}_k\boldsymbol{v}_p\boldsymbol{B}_{ki}I_3(K+k,2K+M+L+m,2K+p) - \sum_{k,l=1}^{K}\boldsymbol{h}_k\boldsymbol{h}_l\boldsymbol{B}_{ki}I_3(K+k,2K+M+L+m,K+l)\right]. \tag{74}$$

D Additional results concerning the SDGM

D.1 SDGM Implementation Details

In the teacher-student model, the multi-head architecture provides a particularly simple measure of feature importance. After Task 1 training, the magnitude of the readout coefficient $|\boldsymbol{h}_i^\dagger|$ quantifies the contribution of hidden unit i to the Task 1 prediction. Since the i-th readout coefficient multiplies the feature generated by the i-th row of the first-layer matrix, hidden units with large $|\boldsymbol{h}_i^\dagger|$ identify feature directions that are most strongly used by Task 1. We therefore rank the hidden units according to their Task 1 readout magnitudes and protect the most important ones during Task 2 adaptation.

Formally, for a given $\kappa \le K$, let $\mathcal{S}_{\text{frozen}} \subset \{1,\dots,K\}$ denote the indices corresponding to the κ largest values of $|\boldsymbol{h}^\dagger|$, so that $|\mathcal{S}_{\text{frozen}}| = \kappa$. The complementary set, $\mathcal{S}_{\text{plastic}} = \{1,\dots,K\}\setminus\mathcal{S}_{\text{frozen}}$, contains the hidden units available for Task 2 adaptation. Importantly, the partition is determined only after Task 1 has been learned and therefore depends on the state reached by the network at the task switch, rather than on a fixed architectural partition specified before training.

We implement SDGM directly within the LoRA parameterization. Rather than optimizing both LoRA factors, we fix the up-projection matrix $\boldsymbol{B}$ to a sparse matrix $\boldsymbol{\Omega} \in \{0,1\}^{K\times L}$ whose non-zero rows are restricted to $\mathcal{S}_{\text{plastic}}$, and optimize only $\boldsymbol{A}$ and $\boldsymbol{h}^\ddagger$. Let $u_1 < \cdots < u_{K-\kappa}$ denote the elements of $\mathcal{S}_{\text{plastic}}$. We assign the L adapter directions cyclically to the plastic units, that is

$$\boldsymbol{\Omega}_{ij} = \begin{cases} 1, & \text{if } i = u_{1+((j-1) \bmod (K-\kappa))}, \\ 0, & \text{otherwise,} \end{cases} \tag{75}$$

Because $\boldsymbol{\Omega}$ is built from the Task 1 state and then held fixed, the same macroscopic theory built for LoRA applies by setting $\boldsymbol{B} = \boldsymbol{\Omega}$ and $\mathrm{d}\boldsymbol{B}/\mathrm{d}\tau = 0$ in Eq. 68, while integrating Eqs.71–74. This allows for a freezing effect. Since $\boldsymbol{h}^\dagger$ is a finite-dimensional parameter tracked explicitly by the theory the partition $\mathcal{S}_{\text{frozen}}$ is itself predicted by the Task 1 ODEs.

Comparison with masked full fine-tuning. For completeness, we also apply the same state-dependent partition to standard full fine-tuning. This comparison can be represented within the same parameterization. By taking $L = K$, setting $\gamma = \sqrt{K}$ and $\boldsymbol{B} = \boldsymbol{I}_K$, we get $\boldsymbol{J} = \boldsymbol{J}_s + \boldsymbol{A}$. With $\boldsymbol{A}$ initialized at zero, optimizing $\boldsymbol{A}$ is equivalent to updating $\boldsymbol{J}$ directly from $\boldsymbol{J}_s$: standard

sequential training is thus totally contained as a sub-case of LoRA fine-tuning. Moreover, replacing $\boldsymbol{I}_K$ by the corresponding diagonal SDGM mask therefore yields masked full fine-tuning as a special case of the same framework.

D.2 Applying the inverse SDGM

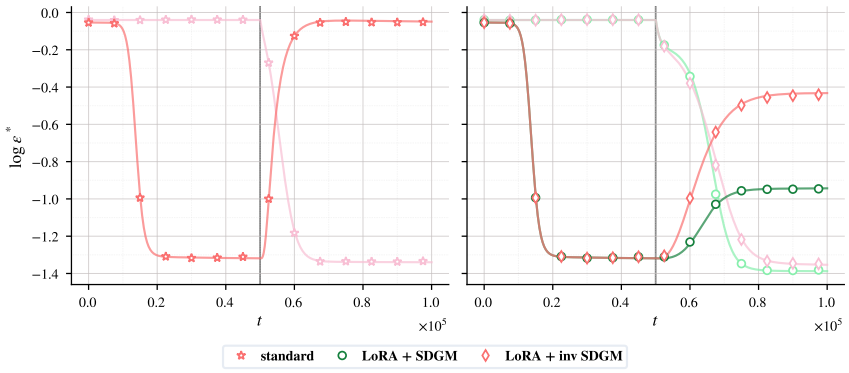

Figure 4: **Ablation of the selection protocol via Inverse SDGM.** Generalization error trajectories on Task 1 and Task 2 under the inverse selection protocol, where student hidden units corresponding to the smallest Task 1 readout magnitudes are frozen during Task 2 training. Comparing this control to standard SDGM disentangles the effect of purely architectural capacity constraints from targeted feature protection. Parameters: $N = 10^3$ $K = 10$, $M = 5$, $L = 5$, $c = 0.5$, $\alpha = 50$, $\kappa = 5$.

To verify that feature selection drives SDGM performance rather than subspace restriction alone, we perform an ablation experiment using an *Inverse SDGM* protocol. In this setting, $\mathcal{S}_{\text{frozen}}$ isolates the κ smallest magnitude entries of $\boldsymbol{h}^{\dagger}$, freezing the least informative directions relative to Task 1. Figure 4 demonstrates that Inverse SDGM yields higher Task 1 generalization error than standard SDGM, establishing that effective feature protection requires explicitly identifying and freezing key task-relevant representations.

D.3 Validation under an unbounded activation function

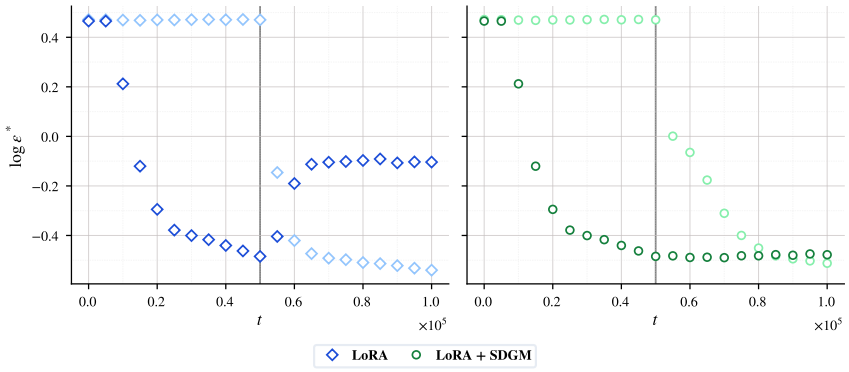

Figure 5: **Validation of SDGM under unbounded ReLU activations.** Generalization error dynamics on Task 1 and Task 2 when both teacher and student networks employ ReLU as the activation function. The lower error on Task 1 confirms that the proposed selection rule mitigates catastrophic forgetting independently of activation saturation. Parameters: $N = 10^3$, $K = 10$, $M = 5$, $L = 5$, $c = 0.5$, $\alpha = 50$. Results shown here are experiment-only.

To verify that the efficacy of SDGM comes from structural information routing rather than artifacts of activation saturation, we evaluate the protocol under an unbounded activation function. Smooth, bounded activations such as $\text{erf}(z)$ naturally constrain preactivation magnitudes. In contrast, the

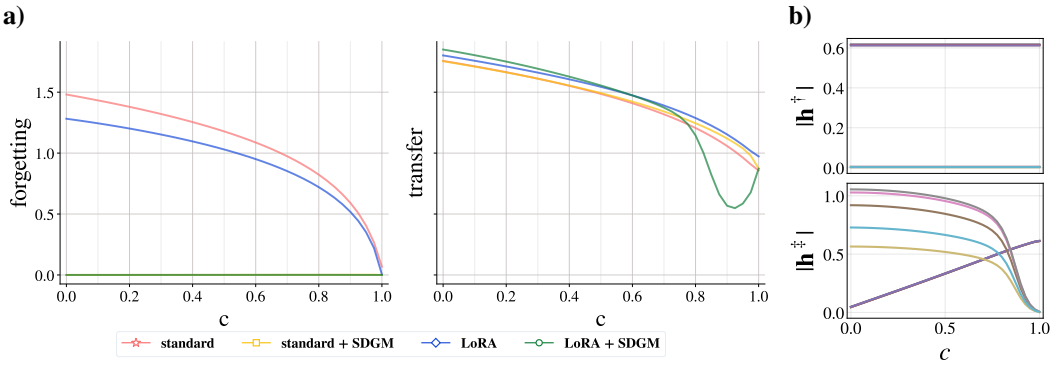

Figure 6: **Typical forgetting on the first task and transfer on the second task with the new initialization**. In this setting, the SDGM procedure allows for no forgetting on the full range of task similarity, while allowing for the same transfer. The smaller transfer at big task similarity for LoRA + SDGM is due to long symmetric plateau, which size increases non-monotonically with c. Parameters : N=10^3, K=10, M=5, L=5, $\alpha = 50$.

Rectified Linear Unit (ReLU), defined as $g(z) = \max(0, z)$, exhibits unbounded values after the first layer. As illustrated in Fig. 5, applying the proposed selection protocol under ReLU dynamics successfully preserves Task 1 performance throughout Task 2 adaptation, with similar transfer on Task 2. This demonstrates that the protocol does not merely exploit head specialization but actively isolates and protects the sub-network carrying critical task representations.

E Results in the specialized regime

All the results presented in this paper are shown in the so-called *symmetric regime*, where the student has not yet been able to specialize towards the specific directions of the teacher. The motivation for this choice of regime is multiple. First, it is more difficult to align with the first task in the *overrealizable regime*, that is when $K > M$. Second, the time constant associated with symmetric subspace escape increases linearly with K. A full discussion on this problematic can be found in Saad & Solla (1995a). Finally, with our choice of readout initialization, the specialization is even more difficult. Multiple works have been done to understand the impact of initialization on forgetting in an equivalent setting, as well as proposing good habits for the initialization scheme (Lee et al., 2022; Jarvis et al., 2025). However, these good habits can be applied with *a priori* knowledge on the tasks that must be fitted, a setting very different from the practitioners experience.
We present here additional results in the *specialized regime*. Following the insights on initialization from Jarvis et al. (2025), we set

$$\boldsymbol{h}_i^\dagger = \begin{cases} 10^{-2} & \text{if } 1 \leq i \leq \lfloor K/2 \rfloor, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \boldsymbol{h}_i^\ddagger = \begin{cases} -1 & \text{if } 1 \leq i \leq \lfloor K/2 \rfloor, \\ 0 & \text{otherwise.} \end{cases}$$

We first check the impact of this new initialization on the readout weights in the unspecialized case in Fig. 6. By artificially forcing the network to only use a fraction of its directions to learn Task 1, SDGM allows for no forgetting on the full range of task similarity. This can be understood by checking the Task 1 readout weights and uncovering that only a fraction of their value is non-0: the initialization biases the network dynamics towards self-pruning, letting free directions for Task 2. At the same time, when training on Task 2, the magnitude of the readout weights associated to new direction decreases monotonically with task similarity. This effect is a consequence of *node re-use* (Lee et al., 2022), where the student is able to recycle directions learned on Task 1, already partially aligned with Task 2. Eventually, when the $c = 1$, the student is not learning any new directions, even after training.

We now turn to the specialized regime, presented in Fig. 7. Even with this initialization, we find that α must be increased to 2000 to observe the exponential decrease in generalization error charac-

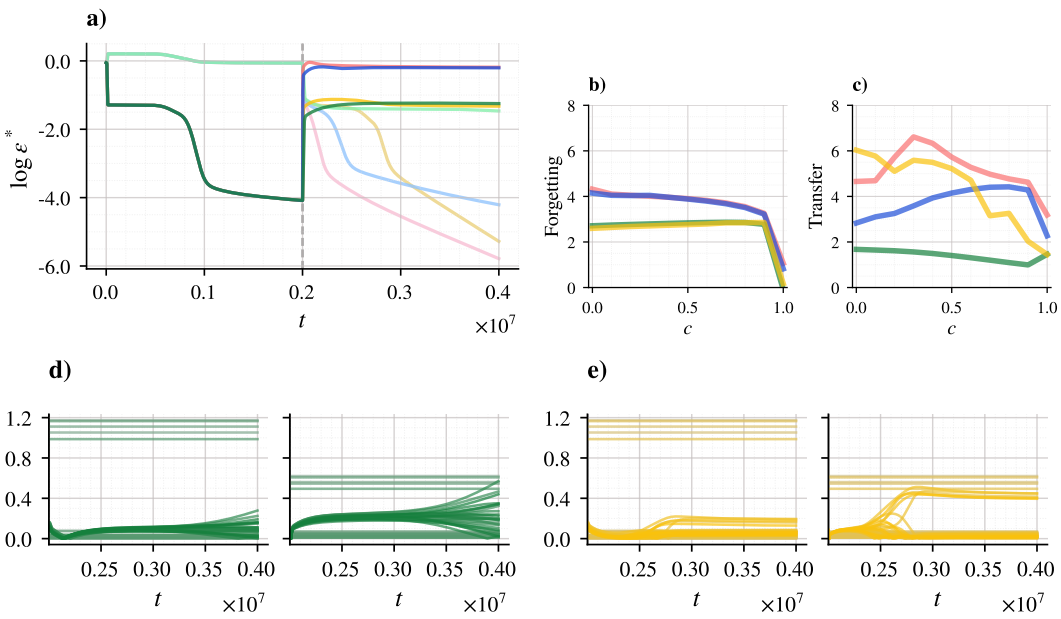

Figure 7: **Additional results in the specialized regimes.** *a)* Typical generalization error for full fine-tuning (red), LoRA (blue), and their SDGM-constrained variants. During Task 1 training, the symmetric plateau, corresponding to the absence of specialization, is located at $\log \epsilon^* \approx -1.5$. *b)* Task 1 forgetting and *c)* Task 2 transfer as a function of teacher similarity c. *d–e)* Student overlaps with the teachers, as defined in Eqs.(18)–(19), after the task switch for LoRA + SDGM (green) and standard + SDGM (yellow). The overlaps corresponding to frozen student directions remain constant throughout Task 2 training.

teristic of specialization. As in the unspecialized regime, LoRA and its SDGM-constrained variant exhibit slower dynamics during Task 2 training, resulting in slower adaptation to the second task. Nevertheless, SDGM enables the student to retain partial alignment with Task 1 while learning Task 2, as shown in Figs. 7b-c. In particular, the prolonged symmetric plateau delays the onset of Task 2 learning, thereby limiting both its acquisition and the subsequent interference with Task 1. In this setting, standard fine-tuning with SDGM adapts more rapidly to Task 2 than its LoRA counterpart, as illustrated in Figs. 7d-e. This faster adaptation leads to greater Task 2 transfer, while the two methods exhibit comparable levels of forgetting.

F Additional details on LoRA

On the initialization of the LoRA matrices In this controlled continual learning setting, the initialization of the LoRA matrices demands careful consideration. The foundational principle of LoRA is to ensure that the weight perturbation is equal to 0 at initialization, that is we force $\Delta \boldsymbol{J} = 0$ when adding the LoRA adapter in order to prevent an immediate disruption of the parameter configuration at the task switch. While the initialization scheme proposed in the seminal LoRA framework (Hu et al., 2021) is tailored to maximize downstream task performance and training stability by letting $\boldsymbol{A}$ start from a Kaiming initialization and setting $\boldsymbol{B} = 0$, our objective introduces a distinct trade-off: we want to achieve high plasticity on Task 2 while keeping stability on Task 1.

Thus, another possible initialization scheme, recently proposed in Rüdiger & Raschka (2026) is to inverse this choice and to let $\boldsymbol{A} = 0$, $\boldsymbol{B} = \mathcal{O}(1)$. This option demonstrated comparable or superior performance across a variety of downstream tasks while mitigating forgetting on Task 1. This mitigation can be understood by first looking at the classical initialization mechanism: initial gradient with respect to $\boldsymbol{A}$ vanishes, leaving the early updates to be driven entirely by the evolution of $\boldsymbol{B}$. In this regime, the random weights of $\boldsymbol{A}$ act as a static random feature projector. This random projection disrupts the alignment between $\boldsymbol{J}_s$ and Task 1. Consequently, the optimization trajectory on Task 2 drives the system into a regime of catastrophic forgetting.

Conversely, the new initialization prevents this destructive mechanism. In this case, the gradient updates of $\boldsymbol{B}$ vanish, forcing $\boldsymbol{A}$ to absorb the initial learning dynamics. The adaptation thus propagates through the low-rank bottleneck L in a more constrained manner, allowing the network to selectively acquire features relevant to Task 2 while maintaining minimal structural overlap with the representation learned for Task 1.

At the same time, randomly selecting the rows of $\boldsymbol{B}$ makes the optimization landscape highly sensitive to initialization, resulting in substantial variability in the trajectory and final configuration of $\boldsymbol{A}$ across random seeds. Within the proposed theoretical framework, this sensitivity is directly visible in the overlaps we recover: since $\boldsymbol{B}$ acts as an order parameter of the system, its initial configuration has a strong impact on the overall training dynamics. To reduce this run-to-run variability, we *initialize* $\boldsymbol{B}$ deterministically as

$$\boldsymbol{B}_{k\ell} = \mathbf{1}\{\ell = 1 + ((k-1) \bmod L)\}\,.$$

This does not alter LoRA's parameterization, for both $\boldsymbol{B}$ and $\boldsymbol{A}$ are trainable low-rank adapters and yields reproducible initial conditions for the corresponding ODE dynamics.

G Trying the various procedures on a real dataset

To validate the predictions of our theory, we apply the proposed procedures to a sequence of simple tasks constructed from the MNIST dataset Lecun et al. (1998). The first task is a binary classification problem in which digits below 5 are assigned to class 0, while digits greater than or equal to 5 are assigned to class 1. The second task uses a different partition of the same dataset, with even digits assigned to class 0 and odd digits to class 1.

We train the model in an online learning setting using the full MNIST dataset. For each digit, the available examples are divided between the two tasks, resulting in $\mu = 3\times10^4$ training examples per task. The 28×28 images are flattened into vectors of dimension $N = 784$. The generalization error curves reported in the main text are averaged over 10 independent training runs, with variability arising from both the data split and the initialization of the student network modules.

We observe the same qualitative behavior as in the theoretical setting: forgetting is largest for the standard procedure and smallest when SDGM is applied to LoRA. The slowdown induced by the LoRA parameterization at the beginning of Task 2 training is also observed in the real-data experiments. The hyper-parameters used are equals to the one used for theoretical simulations, present in Appendix H, the only modification being $N = 784$ in order to match the input size.

H Hyper-parameters for numerical experiments

In this section, we summarize the hyper-parameters that were used to perform all numerical simulations.

- Input dimension: $N = 10^3$,
- Student hidden dimension: $K = 10$,
- Teacher(s) hidden dimension: $M = 5$;
- LoRA rank: $L = 5$ (unless otherwise stated, e.g. Figure 3),
- $\mathrm{teacher}^{\dagger} - \mathrm{teacher}^{\ddagger}$ correlation coefficient: $c = 0.5$ (unless otherwise stated),
- $\mathcal{S}_{\mathrm{frozen}}$ cardinality: $\kappa = K - L$ (unless otherwise stated),
- LoRA prefactor: $\gamma = 1$ (only for LoRA settings),
- Time horizon (for each task): $\alpha = 50$ ($\alpha = 2000$ for the specialized case in Appendix E),

- Learning rates: $\eta_{\boldsymbol{J}} = \eta_{\boldsymbol{B}} = \eta_{\boldsymbol{A}} = \eta_{\boldsymbol{h}} = 0.5$
- Integration step for discretized ODEs using Euler's method: $h_{\text{step}} = 0.05$.

The initialization for teachers and student networks are the following:

- Student first-layer weight $\boldsymbol{J}$: $\boldsymbol{J}_{ij} \sim \mathcal{N}(0, 10^{-6})$,
- Student readouts (for both tasks): $\boldsymbol{h}_i^* \sim \mathcal{N}(0, 10^{-4})$,
- LoRA up-projection adapter $\boldsymbol{B}$: see Appendix F for classical LoRA framework or Equation (75) for LoRA + SDGM,
- LoRA down-projection adapter $\boldsymbol{A}$: $\boldsymbol{A} = 0$,
- $\text{teacher}^\dagger$ first-layer weight $\boldsymbol{W}^\dagger$: $\boldsymbol{W}_{ij}^\dagger \sim \mathcal{N}(0, 1)$;
- $\text{teacher}^\ddagger$ first-layer weight $\boldsymbol{W}^\ddagger$: $\boldsymbol{W}^\ddagger = c\,\boldsymbol{W}^\dagger + \sqrt{1-c^2}\,\mathbf{Z}, \quad \mathbf{Z}_{ij} \sim \mathcal{N}(0, 1)$, with $\mathbf{Z}$ independent of $\boldsymbol{W}^\dagger$,
- Teacher 1 readouts: $\boldsymbol{v}_i^\dagger = +1 + n_i^\dagger, \qquad n_i^\dagger \sim \mathcal{N}(0, 10^{-4})$,
- Teacher 2 readouts $\boldsymbol{v}_i^\ddagger = -1 + n_i^\ddagger, \qquad n_i^\ddagger \sim \mathcal{N}(0, 10^{-4})$.