ICL-TSVD: Bridging Theory and Practice in Continual Learning with Pre-trained Models

Anonymous authors
Paper under double-blind review

Abstract

The goal of continual learning (CL) is to train a model that can solve multiple tasks presented sequentially. Recent CL approaches have achieved strong performance by leveraging large pre-trained models that generalize well to downstream tasks. However, such methods lack theoretical guarantees, making them prone to unexpected failures. Conversely, principled CL approaches often fail to achieve competitive performance. In this work, we bridge this gap between theory and practice by integrating an empirically strong approach (RanPAC) into a principled framework, Ideal Continual Learner (ICL), *designed* to prevent forgetting. Specifically, we lift pre-trained features into a higher dimensional space and formulate an over-parametrized minimum-norm least-squares problem. We find that the lifted features are highly ill-conditioned, potentially leading to large training errors (numerical instability) and increased generalization errors. We address these challenges by continually truncating the singular value decomposition (SVD) of the lifted features. Our approach, termed ICL-TSVD, is stable with respect to the choice of hyperparameters, can handle hundreds of tasks, and outperforms state-of-the-art CL methods on multiple datasets. Importantly, our method satisfies a recurrence relation throughout its continual learning process, which allows us to prove it maintains small training and generalization errors by appropriately truncating a fraction of SVD factors. This results in a stable continual learning method with strong empirical performance and theoretical guarantees.

1 Introduction

Continual learning (CL) requires training a model that performs well on multiple tasks presented sequentially. A primary challenge in CL is acquiring new knowledge without causing *catastrophic forgetting* (i.e., substantial performance degradation on previously learned tasks). The availability of large pre-trained models has recently facilitated advances in CL. As their weights are typically frozen, they provide highly generalizable features that significantly boost performance with little computational overhead (Wang et al., 2022d;a; McDonnell et al., 2023; Zhou et al., 2024b). Pre-trained models also simplify network design, as concatenating a pre-trained model with a shallow trainable network often attains competitive performance (Zhou et al., 2023; McDonnell et al., 2023).

A notable example is *RanPAC* (McDonnell et al., 2023), which lifts pre-trained features into a higher dimensional space and then trains a ridge-regularized linear classifier on the lifted features. Despite its simplicity, RanPAC has established state-of-the-art performance (Zhou et al., 2024a) and, as far as we know, no subsequent work has been shown to consistently outperform it (Ahrens et al., 2024; Prabhu et al., 2024). However, as we will analyze in detail, the performance of RanPAC is unstable and sensitive to the choice of the ridge regularization parameter, which can make it ill-suited for long task sequences. McDonnell et al. (2023) provides limited theoretical guarantees for RanPAC, and a principled understanding of the method is thus imperative to overcome its instability.

On the other hand, the theoretically grounded *Ideal Continual Learner* (ICL) framework (Peng et al., 2023) prevents catastrophic forgetting by design, as it optimizes the current task under the constraint that prior tasks are solved to global optimality. Peng et al. (2023) derived generalization guarantees for ICL, and described possible implementations for linear regression and matrix factorization. Nonetheless, the zero-forgetting promise of ICL has yet to be fully realized in practical implementations, and its applicability in conjunction with pre-trained models remains largely unexplored.

RanPAC and ICL are prime examples that illustrate the gap that prevails in the CL literature (as we review in Appendix H): Theoretically grounded CL methods tend to be impractical (Evron et al., 2022; Peng & Risteski, 2022; Peng et al., 2023; Cai & Diakonikolas, 2024), while highly performant methods involve solving intricate, non-convex training problems, for which deriving informative theoretical guarantees is challenging (Wang et al., 2022b;c;a; Smith et al., 2023; Wang et al., 2023; Jung et al., 2023; Tang et al., 2023; Gao et al., 2024b; Roy et al., 2024; Kim et al., 2024).

Our primary objective is to bridge this gap between theory and practice in the context of CL with pre-trained models. A natural idea towards this goal is to blend RanPAC and ICL, incorporating the complementary strengths of both approaches. More specifically, we adopt the *random ReLU feature* framework as used by RanPAC, and formulate a *minimum-norm least-squares* problem based on ICL to train a linear classifier (cf. Section 2). Yet, random ReLU features are *double-edged*: while they tend to boost performance by increasing feature separability (Telgarsky, 2022; Min et al., 2024), they are also highly ill-conditioned, bringing computational challenges. Indeed, this ill-conditioning makes RanPAC very sensitive to the choice of the ridge regularization parameter; and a naive combination of RanPAC and ICL would inherit this instability. Crucially, we identify that the instability is related to the extremely small singular values that emerge in the spectrum of the pre-trained random ReLU features as more tasks are observed (Section 3). This intriguing finding motivates our method, termed *ICL-TSVD*, which truncates the extremely small singular values (*truncated SVD*) prior to solving the minimum-norm ICL problem (Section 4). ICL-TSVD bridges the gap between theory and practice by delivering stable and strong performance with theoretical guarantees. Concretely:

- We provide a continual implementation of ICL-TSVD to train an over-parameterized linear classifier with highly ill-conditioned features in a numerically stable fashion (Section 4). We show it stabilizes RanPAC and ICL (Figs. 3 and 4), and it is both more scalable and efficient than RanPAC (Fig. 5).
- We derive theoretical guarantees for ICL-TSVD, proving that it has small estimation and generalization errors when a suitable fraction of SVD factors are truncated (Theorems 1 and 2, Section 5). These results stem from a non-trivial recurrence relation that allows us to capture the continual learning dynamics of ICL-TSVD (Lemma 1, Appendix D).
- We conduct extensive experiments on multiple datasets, showing that ICL-TSVD uniformly outperforms prior works and specifically RanPAC (Section 6). Thanks to our stable implementation, ICL-TSVD outmatches RanPAC by a significant margin in the CIL setting with one class given at a time (Inc-1), where hundreds of tasks (classes) are sequentially presented (Table 2).

2 Technical Background

Problem Setting. We consider classification tasks in the *class-incremental learning* (CIL) setting, where each incoming task contains only unseen classes. Following conventions (Yan et al., 2021; Zhou et al., 2023), we write B-q_1, Inc-q_2 to mean that the model is given q_1 classes in the first task and then q_2 classes in each of the subsequent tasks ($q_1 = 0$ means all tasks have q_2 distinct classes). We use *vision transformers* (ViTs) of Dosovitskiy et al. (2021) as pre-trained models.

Pretrained Features and Labels. Given m_t images of task t, we feed them to pre-trained ViTs, obtaining the output features $\boldsymbol{X}_t \in \mathbb{R}^{d \times m_t}$. Here, d is the feature dimension ($d = 768$ in the ViTs used). Corresponding to $\boldsymbol{X}_t$ is the label matrix $\boldsymbol{Y}_t \in \mathbb{R}^{c_t \times m_t}$. Every column of $\boldsymbol{Y}_t$ is a *one-hot vector*, that is some standard basis vector in $\mathbb{R}^{c_t}$, where c_t is the total number of classes observed so far. We thus have $c_1 \leq \cdots \leq c_i \leq \cdots \leq c_t$. Let $M_t := m_1 + \cdots + m_t$. While $\boldsymbol{Y}_i \in \mathbb{R}^{c_i \times m_t}$ might have a different number of rows as c_i varies, one can pad $c_t - c_i$ zero rows to $\boldsymbol{Y}_i$ when new class information is revealed; so, with a slight abuse of notation, $\boldsymbol{Y}_i$ is viewed as having c_t rows. We denote by $\boldsymbol{Y}_{1:t}$ the label matrix of the first t tasks: $\boldsymbol{Y}_{1:t} = [\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_t] \in \mathbb{R}^{c_t \times M_t}$.

Random ReLU Features. Let $\mathrm{relu} : \xi \mapsto \max\{0, \xi\}$ be a ReLU layer and $\boldsymbol{P} \in \mathbb{R}^{E \times d}$ denote a random Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries; here we assume $E \gg d$. These allow us to embed $\boldsymbol{X}_t$ into a higher dimensional space and get *random ReLU features* $\boldsymbol{H}_t \in \mathbb{R}^{E \times m_t}$ via

$$\boldsymbol{H}_t := \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t), \qquad \boldsymbol{H}_{1:t} := [\boldsymbol{H}_1, \ \dots, \boldsymbol{H}_t] \in \mathbb{R}^{E \times M_t}. \tag{1}$$

Note that relu is a pointwise non-linearity, applied to $\boldsymbol{P}\boldsymbol{X}_t$ entry-wise. The goal is to learn a linear classifier $\boldsymbol{W} \in \mathbb{R}^{c_t \times E}$ continually, using features $\boldsymbol{H}_t$ and labels $\boldsymbol{Y}_t$ of task t.

ICL and RanPAC Formulations. With random ReLU features $\boldsymbol{H}_t$ and labels $\boldsymbol{Y}_t$ of task t, we can now instantiate the ICL framework of Peng et al. (2023) as a linearly constrained quadratic problem:

$$\min_{\boldsymbol{W}\in\mathbb{R}^{c_t\times E}} \|\boldsymbol{W}\boldsymbol{H}_t-\boldsymbol{Y}_t\|_{\mathrm{F}}^2 \qquad \text{s.t.} \qquad \boldsymbol{W}\boldsymbol{H}_i=\boldsymbol{Y}_i, \quad i=1,\dots,t-1. \tag{ICL}$$

Since E is assumed to be very large, ICL is always feasible. Moreover, it has infinitely many global minimizers reaching zero loss, and every such minimizer $\boldsymbol{W}$ satisfies $\boldsymbol{W}\boldsymbol{H}_i=\boldsymbol{Y}_i$ for every $i=1,\dots,t$. A popular approach is to select the one with the minimum norm:

$$\min_{\boldsymbol{W}\in\mathbb{R}^{c_t\times E}} \|\boldsymbol{W}\|_{\mathrm{F}}^2 \qquad \text{s.t.} \qquad \boldsymbol{W}\boldsymbol{H}_i=\boldsymbol{Y}_i, \quad i=1,\dots,t. \tag{Min-Norm ICL}$$

Since the constraints $\boldsymbol{W}\boldsymbol{H}_i=\boldsymbol{Y}_i$ force previous tasks to be solved to global optimality, both ICL and Min-Norm ICL prevent forgetting by design. That said, no CL method was implemented for solving Min-Norm ICL in the work of Peng et al. (2023) in the context of pre-trained models.

The RanPAC formulation of McDonnell et al. (2023) is a ridge regression program ($\lambda>0$):

$$\min_{\boldsymbol{W}\in\mathbb{R}^{c_t\times E}} \lambda\cdot\|\boldsymbol{W}\|_{\mathrm{F}}^2+\sum_{i=1}^{t}\|\boldsymbol{W}\boldsymbol{H}_i-\boldsymbol{Y}_i\|_{\mathrm{F}}^2. \tag{RanPAC}$$

The importance of RanPAC lies in its ability to achieve remarkable accuracy across a range of CL datasets when λ is appropriately chosen, as evidenced in Table 1 of Zhou et al. (2024a). That said, no formal analysis of ridge regression was given in the work of McDonnell et al. (2023).

Remark 1. An alternative way to view these formulations is through a two-layer neural network (NN) model, in which pre-trained features $\boldsymbol{X}$ are fed to a NN

$$\boldsymbol{X}\mapsto\boldsymbol{W}\cdot\mathrm{relu}(\boldsymbol{P}\boldsymbol{X}),$$

where $\boldsymbol{P}$ are randomly generated, then fixed, and $\boldsymbol{W}$ are trainable parameters. On a historical note, networks of this form predate the early work of Schmidt et al. (1992); Pao & Takefuji (1992). The latter are known as *extreme learning machines* and *random feature models*. In Appendix H.2 we review these lines of research, providing connections to our work.

Remark 2. The MSE loss is used for ICL and RanPAC, and will also be used in our approach (Section 4). Janocha & Czarnecki (2017); Hui & Belkin (2021) showed that, in many settings, the MSE and cross-entropy losses yield similar performance. We utilize the MSE loss as it allows for a closed-form least-squares solution to be rapidly computed and continually updated.

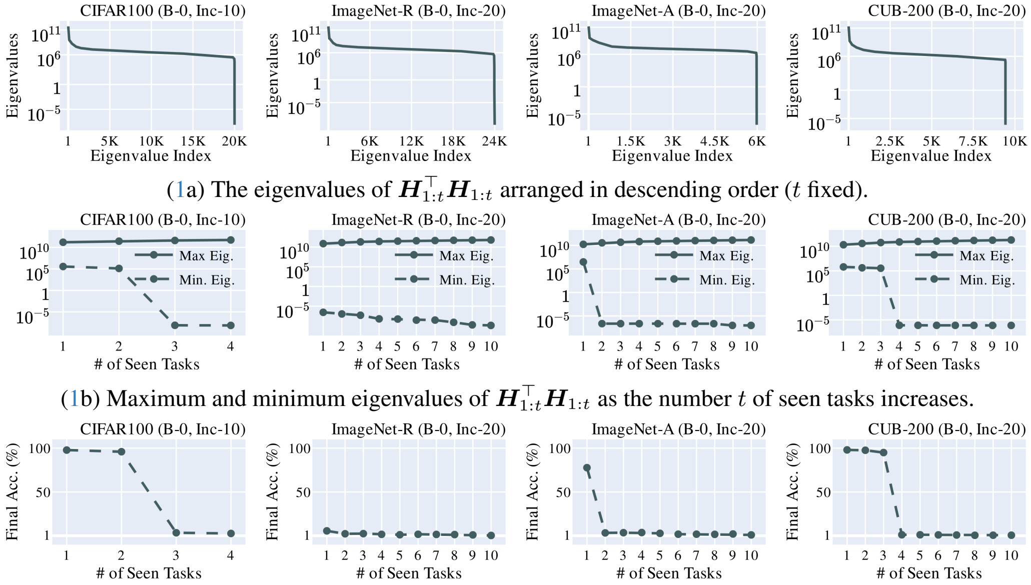
Remark 3. In RanPAC, *first-session adaptation* is executed before the first CL iteration. That is, the pre-trained model is fine-tuned with data from the first task in a *parameter-efficient* way (Panos et al., 2023). This needs extra hyperparameters and yields different features than $\boldsymbol{H}_{1:t}$. We study the impact of this step in Table 1, Section 6, and Table 11, Appendix K.2.

3 Emergence of Instability

In this section, we showcase and analyze the instability of Min-Norm ICL and RanPAC.

Instability of Min-Norm ICL. Fig. 1a plots the eigenvalues of $\boldsymbol{H}_{1:t}^\top\boldsymbol{H}_{1:t}$, showing that $\boldsymbol{H}_{1:t}^\top\boldsymbol{H}_{1:t}$ is highly ill-conditioned and has just a few extremely large and extremely small eigenvalues, outnumbered by the eigenvalues in between that decay more slowly. Fig. 1b plots extreme eigenvalues of $\boldsymbol{H}_{1:t}^\top\boldsymbol{H}_{1:t}$, revealing that the minimum eigenvalue drastically drops after a certain number of tasks. Comparing Fig. 1b and Fig. 1c, we see that the accuracy of solving Min-Norm ICL plummets exactly when the extremely small eigenvalues emerge and begin to invade the spectrum.

Instability of RanPAC. Fig. 2 shows RanPAC with a small regularization parameter λ can fail to achieve competitive performance, while it seems to work well with a large enough λ (e.g., $\lambda=10^4$). In constrast, Prabhu et al. (2024) finds that small λ (of order 10^{-5}) works better when $\boldsymbol{H}_{1:t}$ is replaced with random Fourier features. This suggests that the optimal choice of λ depends, among other factors, on the scale of the features and the noise level. McDonnell et al. (2023) selects λ from the pool of candidates $\{10^{-8},10^{-7},\dots,10^{8}\}$ via cross-validation on a small faction of training data. Although this suffices to stabilize RanPAC in some settings, cross-validation can fail when the validation (or training) set of the current task is small and not representative of test data. Unfortunately, this failure occurs often in CIL with small increments (cf. Section 6).



(1a) The eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ arranged in descending order (t fixed).

(1b) Maximum and minimum eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ as the number t of seen tasks increases.

(1c) Final test accuracy of the incremental SVD solution to Min-Norm ICL (subject to numerical errors).

Figure 1: Spectral analysis of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ and its impact on test accuracy (see also Appendix K.5). The matrix $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ is highly ill-conditioned (1a); and test accuracy drops drastically when small eigenvalues (of order 10^{-5}) invade the spectrum (1b, 1c).

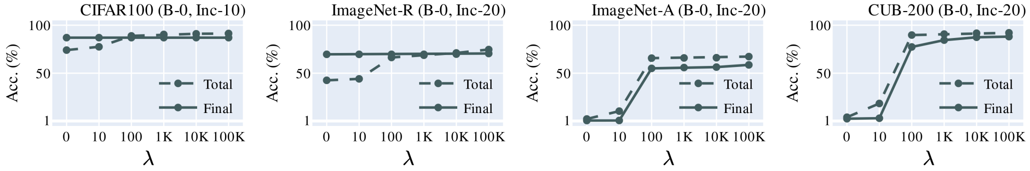


Figure 2: RanPAC is unstable as it breaks down for the small regularization λ. Cross-validation does not fully stabilize RanPAC (cf. Section 6). See Section 6 for detailed experimental settings.

The Cause of Instablity. Two types of errors could give rise to instability of Min-Norm ICL and RanPAC. First of all, the fact that $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ is highly ill-conditioned (Fig. 1a) can lead to numerical errors. Also, extremely small eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ can account for generalization errors (Schaeffer et al., 2024). In the case of Min-Norm ICL, the training MSE loss $\frac{1}{M_t}\|\boldsymbol{W}\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\|_{\mathrm{F}}^2$ of the solution to Min-Norm ICL explodes (Fig. 3) precisely when the extremely small eigenvalues emerge (Fig. 1b). Thus, we argue that numerical errors are the main cause underlying the instability of Min-Norm ICL. Since RanPAC does not suffer from this training loss explosion (cf. Appendix K.7.1), its instability is more likely due to generalization errors.

4 ICL-TSVD: Stabilizing ICL via Truncated SVD

Offline Description. The numerical evidence collected so far suggests that the instability of Min-Norm ICL relates to the emergence of extremely small eigenvalues. This motivates a simple remedy, called *ICL-TSVD* (*offline description*), which consists in truncating the smallest singular values (vectors) of $\boldsymbol{H}_{1:t}$ and then solving Min-Norm ICL using the truncated version of $\boldsymbol{H}_{1:t}$. More concretely, write the SVD of $\boldsymbol{H}_{1:t}$ as $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \cdots + \sigma_{M_t} \boldsymbol{u}_{M_t} \boldsymbol{v}_{M_t}^\top$ with ordered singular values $\sigma_1 \geq \cdots \geq \sigma_{M_t}$. The truncation can then be described with some integer $k_t \in [1, M_t]$ by a function τ_{k_t} that maps $\boldsymbol{H}_{1:t}$ to $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \cdots + \sigma_{k_t} \boldsymbol{u}_{k_t} \boldsymbol{v}_{k_t}^\top$; here k_t is the number of top SVD factors preserved. Applying this idea to Min-Norm ICL means solving the following program:

$$\overline{\boldsymbol{W}}_t \in \operatorname*{argmin}_{\boldsymbol{W} \in \mathbb{R}^{c_t \times E}} \|\boldsymbol{W}\|_{\mathrm{F}}^2 \qquad \text{s.t.} \qquad \boldsymbol{W}\tau_{k_t}(\boldsymbol{H}_{1:t}) = \boldsymbol{Y}_{1:t}. \tag{ICL-TSVD}$$

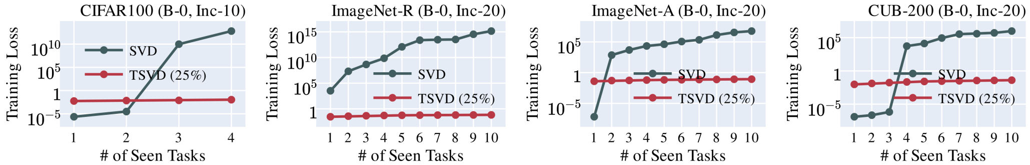


Figure 3: The average training MSE loss $\frac{1}{M_t}\|\boldsymbol{W}\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\|_{\mathrm{F}}^2$ of the incremental SVD solution to Min-Norm ICL explodes when extremely small eigenvalues emerge (Fig. 1b). TSVD (25%) truncates 25% minimum singular values and implements ICL-TSVD online, stabilizing Min-Norm ICL.

Algorithm 1: Continual Solver of ICL-TSVD (detailed version in Algorithm 4)

```
1  Input (Task t): Features $\boldsymbol{H}_t \in \mathbb{R}^{E\times m_t}$ (1), labels $\boldsymbol{Y}_t \in \mathbb{R}^{c_t\times m_t}$, truncation percentage $\zeta \in [0,1]$;
2  For $t \leftarrow 1, 2, \dots$:
3      $k_t \leftarrow \lceil (1-\zeta)\min\{E, M_t\}\rceil$;    // $M_t := m_1 + \cdots + m_t$ can be updated online
4      $\boldsymbol{J}_{1:t} \leftarrow \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$;    // online update of $\boldsymbol{J}_{1:t}$ detailed in Algorithm 5, Appendix C
5      Form $\widetilde{\boldsymbol{B}}_t$ as per (3);
6      $(\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}) \leftarrow$ Top-$k_t$ SVD factors of $\widetilde{\boldsymbol{B}}_t$;    // Algorithm 3 if $t = 1$, or Algorithm 2 if $t > 1$
7      Compute linear classifier $\widetilde{\boldsymbol{W}}_t := \boldsymbol{J}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$;    // cf. (2) and (4)
```

It will be useful to write down the closed-form expression of $\overline{\boldsymbol{W}}_t$. Let $\boldsymbol{U}_{1:t}\boldsymbol{\Sigma}_{1:t}\boldsymbol{V}_{1:t}^\top$ be an SVD of $\boldsymbol{H}_{1:t}$, and $\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}\overline{\boldsymbol{V}}_{1:t}^\top$ an SVD of $\tau_{k_t}(\boldsymbol{H}_{1:t})$, where $\boldsymbol{\Sigma}_{1:t}$, $\overline{\boldsymbol{\Sigma}}_{1:t}$ are invertible square matrices. Then

$$\begin{aligned}\overline{\boldsymbol{W}}_t &= \boldsymbol{Y}_{1:t}\overline{\boldsymbol{V}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-1}\overline{\boldsymbol{U}}_{1:t}^\top = \boldsymbol{Y}_{1:t}\overline{\boldsymbol{V}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top\left(\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top\right)\\ &\overset{\text{(i)}}{=} \boldsymbol{Y}_{1:t}\boldsymbol{V}_{1:t}\boldsymbol{\Sigma}_{1:t}\boldsymbol{U}_{1:t}^\top\left(\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top\right) = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\left(\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top\right),\end{aligned} \tag{2}$$

where equality (i) holds as the column vectors of $\boldsymbol{U}_{1:t}$ not shown in $\overline{\boldsymbol{U}}_{1:t}$ are orthogonal to $\overline{\boldsymbol{U}}_{1:t}$.

Remark 4. ICL-TSVD combines *principal component analysis* and *ordinary least-squares*, which is analogous to *principal component regression* (PCR) (Xu & Hsu, 2019; Huang et al., 2022; Hucker & Wahl, 2023; Bach, 2024; Green & Romanov, 2024). While PCR truncates the SVD of $\boldsymbol{H}_{1:t}$ only once, our implementation of ICL-TSVD performs the truncation continually for each t. Hence, our approach can be viewed as *continual PCR*. See also Remark 6.

Continual Implementation. In order to turn (2) into a continual implementation, we need to update $\boldsymbol{J}_{1:t} := \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ and $\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top$ in an online fashion. This procedure is described in Algorithm 1, where the following points are considered:

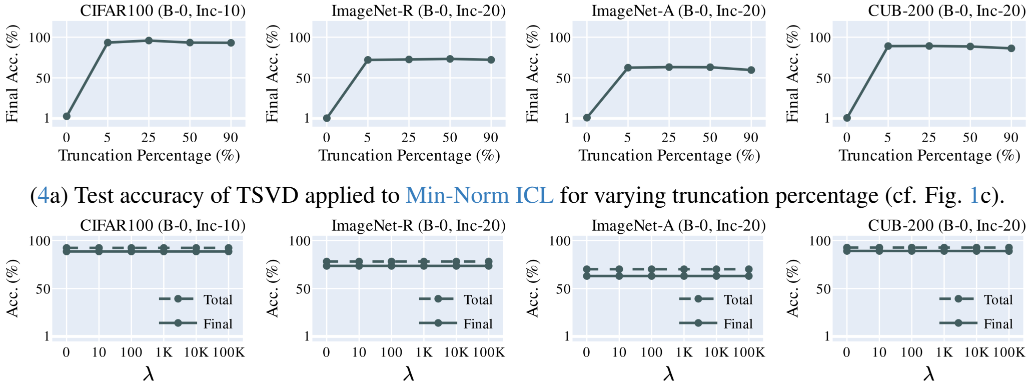
- Since the columns of $\boldsymbol{Y}_{1:t}$ are one-hot vectors, we can compute $\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ incrementally by matrix addition rather than (sparse) matrix multiplication.

- An exact update of $\overline{\boldsymbol{U}}_{1:t}$ and $\overline{\boldsymbol{\Sigma}}_{1:t}$ would require computing the SVD factors of the full data $\boldsymbol{H}_{1:t}$. However, past data $\boldsymbol{H}_{1:t-1}$ is not available when observing task t. Thus, for each task t, we maintain two matrices $\widetilde{\boldsymbol{U}}_{1:t}$, $\widetilde{\boldsymbol{\Sigma}}_{1:t}$; which capture the information in $\overline{\boldsymbol{U}}_{1:t}$ and $\overline{\boldsymbol{\Sigma}}_{1:t}$. Specifically, as shown in Line 6, we set $\widetilde{\boldsymbol{U}}_{1:t}$, $\widetilde{\boldsymbol{\Sigma}}_{1:t}$ to be the top k_t SVD factors of $\widetilde{\boldsymbol{B}}_t$, where $\widetilde{\boldsymbol{B}}_t$ is defined as

$$\widetilde{\boldsymbol{B}}_t := \begin{cases}\boldsymbol{H}_1 & \text{if } t = 1;\\ [\widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{\Sigma}}_{1:t-1},\ \boldsymbol{H}_t] & \text{otherwise.}\end{cases} \tag{3}$$

The top-k_t singular values of $\widetilde{\boldsymbol{B}}_t$ and $\boldsymbol{H}_{1:t}$ are close to each other (cf. Fig. 1a, Fig. 9, Fig. 10, and Theorem 6), which showcases the effectiveness of our continual updating strategy. Then, as shown in Line 7 and recalling (2) and $\boldsymbol{J}_{1:t} := \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$, we construct a linear classifier via

$$\widetilde{\boldsymbol{W}}_t \leftarrow \boldsymbol{J}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top. \tag{4}$$



(4a) Test accuracy of TSVD applied to Min-Norm ICL for varying truncation percentage (cf. Fig. 1c).

(4b) Test accuracy of TSVD (25%) applied to RanPAC for varying regularization parameter λ (cf. Fig. 2).

Figure 4: TSVD stabilizes Min-Norm ICL and RanPAC ($E = 10^5$).

- In Algorithm 1, ζ denotes the truncation percentage. Given ζ, we set the number k_t of top SVD factors preserved at task t to $k_t = \lceil (1-\zeta)\min\{E, M_t\} \rceil$, where $\lceil \cdot \rceil$ converts an input number to the closest integer no smaller than it. Note that the choice of ζ is not sensitive to the scale of the data, unlike the ridge regularization parameter λ of RanPAC (recall Section 3).

Remark 5. At test time, given a sample, we make a forward pass and obtain its random ReLU feature $\boldsymbol{h}$. The predicted class is then set according to the maximum entry of $\widetilde{\boldsymbol{W}}_t \boldsymbol{h}$.

Stability Improvements. Our method stabilizes the training losses (Fig. 3) and stabilizes Min-Norm ICL for a wide range of truncation percentages (Fig. 4a). Furthermore, we extend the solution of (4) into $\boldsymbol{J}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}(\widetilde{\boldsymbol{\Sigma}}_{1:t}^2 + \lambda \boldsymbol{I}_E)^{-1}\widetilde{\boldsymbol{U}}_{1:t}^\top$ for ridge regression, where $\boldsymbol{I}_E$ is the $E \times E$ identity matrix. This stabilizes RanPAC and is practically immune to changes in the regularization parameter λ (Fig. 4b). Extensive experimental validation is provided in Section 6 and Appendix K.

5 Provably Controlled Estimation and Generalization

In this section, we present Theorems 1 and 2, which bound the estimation and generalization error of the output (4) of our approach (Algorithm 1).

Notations. Denote by $\mu_k(\cdot)$ the k-th largest eigenvalue of a symmetric matrix. Let

$$\gamma_1 = 1, \quad \gamma_t := \frac{\mu_{k_t}\big(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\big)}{\max_{i=1,\dots,t-1}\Big\{\mu_{k_i+1}\big(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\big)\Big\}}, \quad \forall\, t > 1. \tag{5}$$

The quantity γ_t relates to the *stability-plasticity tradeoff*, as it is the ratio between the *minimum preserved eigenvalue* $\mu_{k_t}\big(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\big)$ at task t and the *maximum eigenvalues being truncated in the past*, $\mu_{k_i+1}\big(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\big)$. Clearly $\gamma_t > 0$, as we truncate only non-zero eigenvalues. Furthermore, if we truncate eigenvalues smaller than a given threshold δ, we have $\gamma_t \geq 1$. In this case, the threshold δ implicitly determines k_i's and we have $\mu_{k_t}\big(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\big) \geq \delta > \mu_{k_i+1}\big(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\big)$. Finally, as suggested by Fig. 1a, γ_t can be as large as 10^{10}. For instance, if we set $\delta = 10^{-2}$, then the maximum truncated eigenvalue is of order 10^{-5} and the minimum preserved is of order 10^5.

Then, the *accumulative error* a_t is defined as

$$a_0 = 0, \qquad a_t := \sum_{i=1}^{t} \mu_{k_i+1}\big(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\big), \quad \forall t \geq 1. \tag{6}$$

The term a_t reflects the information ignored by our algorithm, as $\mu_{k_i+1}\big(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\big)$ is the maximum eigenvalue truncated at task i. Note that, even when observing thousands of tasks (e.g., $t \approx 10^3$), if we truncate the smallest eigenvalues (of order 10^{-5}), a_t is in the order of 10^{-2}.

Model Assumption. We consider a noisy linear regression model. Specifically, we assume there is some *ground-truth* weight matrix $\boldsymbol{W}_t^* \in \mathbb{R}^{c_t \times E}$ and noise $\mathcal{E}_{1:t} \in \mathbb{R}^{c_t \times M_t}$ satisfying

$$\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}. \tag{7}$$

The quantities $\boldsymbol{W}_t^*$ and $\mathcal{E}_{1:t}$ are colored to reflect the fact that they are unknown and not computable. The model in (7) is related to *probabilistic principal component analysis* (PPCA); cf. Tipping & Bishop (1999) and Chapter 2.2 of Vidal et al. (2016). The two main differences with PPCA are that we make no probabilistic assumptions on $\boldsymbol{H}_{1:t}$ or $\mathcal{E}_{1:t}$ (except in Appendix F); and we consider the overparameterized case with large E, while PPCA assumes $\boldsymbol{W}_t^*$ is a tall matrix (i.e., $E < c_t$).

Estimation Guarantee. In the over-parametrized regime $E \gg M_t$, a solution to Min-Norm ICL should, in principle, perfectly fit the data and achieve zero training MSE. However, solving Min-Norm ICL is numerically unstable and empirically entails huge losses (Fig. 3). As a remedy, our approach truncates the data spectrum continually, trading-off between perfectly fitting training data and increasing numerical stability. The following theorem, whose proof can be found in Appendix E, connects the eigenvalue ratio γ_t and the accumulative error a_t with our method's training loss, showing that the estimation error is provably under control:

Theorem 1. *Let $\widetilde{\boldsymbol{B}}_t, \gamma_t, a_t$ be defined as in (3), (5), and (6) respectively. If $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}$ (7), then the output $\widetilde{\boldsymbol{W}}_t = \boldsymbol{Y}_{1:t} \boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top$ of our method (4) satisfies*

$$\begin{aligned} \frac{1}{M_t} \left\| \widetilde{\boldsymbol{W}}_t \boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t} \right\|_{\mathrm{F}}^2 \leq 4 \cdot \| \boldsymbol{W}_t^* \|_{\mathrm{F}}^2 \left(\frac{a_t}{M_t} + \frac{a_{t-1}(t-1)}{\gamma_t M_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2 M_t} \right) \\ + 2 \cdot \| \mathcal{E}_{1:t} \|^2 \left(\frac{(M_t - k_t)}{M_t} + \frac{(t-1) \min \{ M_{t-1} - k_{t-1}, (t-1) k_t \}}{\gamma_t^2 M_t} \right). \end{aligned} \tag{8}$$

One of the main quantities governing the bound in Theorem 1 is a_t / M_t, which reflects the truncation process for the current task. When truncating the extremely small eigenvalues (of order 10^{-5}) and observing hundreds of tasks, a_t is in the order of 10^{-3}, which makes a_t / M_t insignificant. Then, the terms $(t-1)/\gamma_t$ and a_{t-1}/M_t capture the continual past truncations and are equal to zero for $t = 1$. Similarly to a_t, when truncating only the smallest eigenvalues, we have $a_{t-1} \approx 10^{-5}(t-1)$ and $(t-1)/\gamma_t \approx 10^{-10}(t-1)$. Hence, all terms involving $(t-1)/\gamma_t$ and a_{t-1}/M_t are under control for hundreds- even thousands- of tasks. Finally, although the ground-truth $\boldsymbol{W}_t^*$ and noise $\mathcal{E}_{1:t}$ are unknown, we empirically verify that the minimum-norm solution to ICL-TSVD achieves high accuracy (Section 6). This suggests the linear model assumption is adequate, and that $\|\mathcal{E}_{1:t}\|^2$ and $\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2$ are reasonably small. In summary, the upper bound (8) shown in Theorem 1 behaves well and is quite small if we truncate the eigenvalues suitably (which makes γ_t large and a_t small).

Generalization Guarantee. Consider a test sample $(\boldsymbol{h}, \boldsymbol{y})$ satisfying $\boldsymbol{y} = \boldsymbol{W}_t^* \boldsymbol{h} + \boldsymbol{\epsilon}$ for some noise vector $\boldsymbol{\epsilon}$. To obtain a generalization guarantee that is widely applicable, we only assume that $\boldsymbol{h}$ is randomly sampled from some distribution with a finite second-order moment ($\boldsymbol{\Lambda} := \mathbb{E}[\boldsymbol{h}\boldsymbol{h}^\top] < \infty$), and that $\boldsymbol{\epsilon}$ is random, independent of $\boldsymbol{h}$. Given the output $\widetilde{\boldsymbol{W}}_t$ of our method (4), we bound its test error $\mathbb{E}_{\boldsymbol{h},\boldsymbol{\epsilon}}\left[\|\widetilde{\boldsymbol{W}}_t \boldsymbol{h} - \boldsymbol{y}\|^2\right]$ over the randomness of $\boldsymbol{h}, \boldsymbol{\epsilon}$ as follows:

Theorem 2. *Let $\widetilde{\boldsymbol{B}}_t, \gamma_t, a_t$ be defined as in (3), (5), and (6) respectively. Assume $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}$ (7) and $\boldsymbol{y} = \boldsymbol{W}_t^* \boldsymbol{h} + \boldsymbol{\epsilon}$ with $\boldsymbol{\Lambda} := \mathbb{E}[\boldsymbol{h}\boldsymbol{h}^\top]$. The output $\widetilde{\boldsymbol{W}}_t$ of Algorithm 1 satisfies*

$$\mathbb{E}_{\boldsymbol{h},\boldsymbol{\epsilon}} \left\| \widetilde{\boldsymbol{W}}_t \boldsymbol{h} - \boldsymbol{y} \right\|^2 \leq 4 \cdot \| \boldsymbol{W}_t^* \|_{\mathrm{F}}^2 \cdot \mathbb{B}_t + 4 \cdot \| \mathcal{E}_{1:t} \|^2 \cdot \mathbb{V}_t + 2 \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \left[\| \boldsymbol{\epsilon} \|^2 \right], \tag{9}$$

where $\mathbb{B}_t$ and $\mathbb{V}_t$ are defined as follows:

$$\begin{aligned} \mathbb{B}_t &= \left\| \boldsymbol{\Lambda} - \frac{1}{M_t} \boldsymbol{H}_{1:t} \boldsymbol{H}_{1:t}^\top \right\| \left(1 + \frac{(t-1)^2}{\gamma_t^2} \right) + \left(\frac{a_t}{M_t} + \frac{a_{t-1}(t-1)}{\gamma_t M_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2 M_t} \right) \\ \mathbb{V}_t &= \left\| \boldsymbol{\Lambda} - \frac{1}{M_t} \boldsymbol{H}_{1:t} \boldsymbol{H}_{1:t}^\top \right\| \cdot \frac{\left(\frac{1}{\gamma_t} \min \{ M_{t-1} - k_{t-1}, (t-1) k_t \} + k_t \right)}{\mu_{k_t} (\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top)} \\ &\quad + \frac{k_t}{M_t} + \left(\frac{t-1}{\gamma_t^2 M_t} + \frac{2}{\gamma_t M_t} \right) \cdot \min \{ M_{t-1} - k_{t-1}, (t-1) k_t \}. \end{aligned} \tag{10}$$

There are two major terms in $\mathbb{B}_t$ (10). The term in the right-most large parenthesis also appears in the estimation bound of Theorem 1; and reflects the fact that training losses impact generalization errors. Then, the term $\left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\|$ is commonly seen in *covariance estimation* (Wainwright, 2019), where $\boldsymbol{h}$ and the columns of $\boldsymbol{H}_{1:t}$ are assumed to be independent i.i.d. Gaussian vectors. In this case, if $\boldsymbol{\Lambda}$ furthermore satisfies some boundedness condition, we can show $\left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\|$ converges to 0 as $M_t \to \infty$; cf. Theorem 9 of Koltchinskii & Lounici (2017). Note that $\mathbb{V}_t$ is independent of noise, so the rest of the terms in (9), which are weighted by noise magnitudes $\|\mathcal{E}_{1:t}\|^2$, $\mathbb{E}_{\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon}\|^2\right]$, are negligible if the noise is sufficiently small (as suggested by Section 6).

Remark 6. There is a comprehensive line of works on the generalization theory of linear models in the over-parameterized setting, developed to understand the phenomena such as *benign overfitting* or *double descent* (cf. Appendix H.2). Specifically, our results are related to Xu & Hsu (2019); Huang et al. (2022); Hucker & Wahl (2023); Bach (2024); Green & Romanov (2024), that analyze the generalization errors of principal component regression (cf. Remark 4). Nevertheless, these papers consider the offline setting, where truncation is performed only once. In constrast, we analyze the continual truncation setting, which is most pertinent for CL. Indeed, for $t = 1$, $\mathbb{B}_1$ is equal to the corresponding term in Theorem 1 of Huang et al. (2022) up to a constant. More importantly, these papers make statistical (e.g., Gaussian) assumptions on $\boldsymbol{H}_{1:t}$, which are potentially violated by generating $\boldsymbol{H}_{1:t}$ via $\boldsymbol{H}_t := \operatorname{relu}(\boldsymbol{P}\boldsymbol{X}_t)$, with $\boldsymbol{X}_t$ consisting of features from pre-trained models. In contrast, Theorems 1 and 2 have few assumptions, and so they apply, at least in principle, to the *full* architecture (i.e., a pre-trained model and random ReLU feature model in cascade).

6 Numerical Validation

This section highlights the performance and efficiency of ICL-TSVD in the CIL setting across a diverse range of datasets and increments. For additional results, see Appendix K, particularly Appendix K.3 for experimental outcomes in the DIL (*domain-incremental learning*) setting.

6.1 Setup

Baselines. The most relevant baseline to compare is RanPAC (McDonnell et al., 2023). Additional competitive baselines include L2P (Wang et al., 2022d), DualPrompt (Wang et al., 2022c), CodaPrompt (Smith et al., 2023), SimpleCIL, ADAM (Zhou et al., 2023) and EASE (Zhou et al., 2024b). We also compare ICL-TSVD with a *joint linear classifier*, that is, a linear model trained using either the pre-trained features $\boldsymbol{X}_{1:T}$ of all T tasks, or the random ReLU features $\boldsymbol{H}_{1:T}$. We denote these two methods by LC ($\boldsymbol{X}_{1:T}$) and LC ($\boldsymbol{H}_{1:T}$). To ensure a fair comparison, all experiments are conducted based on the PILOT GitHub repository of Sun et al. (2023). Additional experimental details, as well as a comprehensive review of relevant baselines is given in Appendix J and Appendix H.

Pre-trained Models. We use ViT models pre-trained on ImageNet-1K; specifically the model `vit_base_patch16_224` from the `timm` repository (Wightman, 2019). Experiments using ViTs pre-trained on ImageNet-21K are presented in Appendix K.2.

Datasets. Following prior works (Zhou et al., 2023; McDonnell et al., 2023), we run CIL experiments with B-q_1, Inc-q_2 on continual learning versions of the following datasets: CIFAR100 (Krizhevsky et al., 2009), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al., 2021b), CUB-200 (Wah et al., 2011), ObjectNet (Barbu et al., 2019), OmniBenchmark (Zhang et al., 2022), VTAB (Zhai et al., 2019), and StanfordCars (Krause et al., 2013). We set $q_1 = 0$ for most cases, but since StanfordCars and VTAB have 196 and 50 classes, respectively, we take $q_1 = 16$ and $q_1 = 10$ for them. We let q_2 vary in $\{5, 10, 20\}$, and also consider the more challenging case $q_2 = 1$.

Metrics. After learning task t we evaluate the top-1 classification accuracy $\mathcal{A}_{i,t}$ for every $i = 1, \dots, t$. For a total of T tasks, the *accuracy matrix* $\mathcal{A}$ is defined as a $T \times T$ upper triangular matrix with its (i, t)-th entry being $\mathcal{A}_{i,t}$. *Final accuracy* is defined as the average $\frac{1}{T}\sum_{i=1}^{T}\mathcal{A}_{i,T}$ of the last column of $\mathcal{A}$. *Total accuracy* is defined as the average $\frac{1}{T(T-1)}\sum_{1\leq i\leq t\leq T}\mathcal{A}_{i,t}$ of all upper triangular entries. Following common practices, we use total accuracy and final accuracy as our evaluation metrics.

Table 1: Final accuracy with pre-trained ViTs. Large accuracy gaps between RanPAC and ICL-TSVD (ours) are shown in bold. †: Methods using first-session adaptation with the corresponding hyperparameters set as per RanPAC† (cf. Remark 3). $*$: Methods using first-session adaptation with the hyperparameters set as per EASE* (Zhou et al., 2024b). Details are in Appendix J.

(Part 1)	CIFAR100 (B-0)			ImageNet-R (B-0)			ImageNet-A (B-0)			CUB-200 (B-0)			Avg.
LC ($\boldsymbol{X}_{1:T}$)	87.56			72.42			58.85			88.76			76.90
LC ($\boldsymbol{H}_{1:T}$)	87.76			73.00			59.25			88.72			77.18
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
L2P	80.25	83.53	83.57	67.92	71.78	73.42	44.50	48.52	51.28	53.60	59.20	67.81	65.45
DualPrompt	80.85	83.86	84.59	67.12	71.57	72.87	49.70	53.72	56.75	54.79	63.99	69.93	67.48
CodaPrompt	82.93	86.31	87.87	67.80	72.73	74.85	34.43	49.57	59.51	36.39	60.18	71.29	65.32
SimpleCIL	80.48	80.48	80.48	63.47	63.47	63.47	58.72	58.72	58.72	80.45	80.45	80.45	70.78
RanPAC	86.71	87.02	87.10	71.90	71.97	72.50	**56.48**	62.34	61.75	88.08	87.15	88.13	76.76
ICL-TSVD	88.18	88.18	88.21	73.67	73.72	73.63	**62.74**	63.20	63.20	89.36	89.27	89.23	78.55
ADAM†	83.55	85.13	85.86	63.73	65.03	71.40	58.72	58.66	58.99	80.49	80.66	81.00	72.77
RanPAC†	88.73	90.04	90.74	70.80	73.37	78.80	62.34	62.08	62.28	88.42	87.57	88.68	78.65
ICL-TSVD†	89.73	90.82	91.44	73.58	74.55	79.13	62.74	62.80	62.94	89.14	89.19	89.27	79.61
EASE*	84.43	86.48	88.16	73.53	77.02	77.55	58.26	61.69	62.28	80.66	81.68	81.13	76.07
ICL-TSVD*	89.46	90.90	91.67	78.73	80.43	81.45	63.40	64.45	65.64	89.14	89.19	89.44	81.16
(Part 2)	ObjectNet (B-0)			OmniBenchmark (B-0)			VTAB (B-10)			StanfordCars (B-16)			Avg.
LC ($\boldsymbol{X}_{1:T}$)	59.70			79.55			91.32			74.12			76.17
LC ($\boldsymbol{H}_{1:T}$)	59.96			80.02			91.17			73.65			76.20
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
L2P	45.53	52.05	55.49	54.50	57.29	60.50	59.32	73.25	78.91	13.70	27.46	43.68	51.81
DualPrompt	47.56	53.68	55.64	56.14	59.18	62.39	64.10	77.78	83.75	11.38	18.84	27.89	51.53
CodaPrompt	46.61	54.44	59.17	60.00	64.98	68.25	68.77	76.81	86.32	7.96	11.29	30.74	52.95
SimpleCIL	51.66	51.66	51.66	70.19	70.19	70.19	82.53	82.53	82.53	35.46	35.46	35.46	59.96
RanPAC	58.77	57.66	57.69	77.63	77.63	77.46	91.15	91.58	91.58	**58.03**	71.40	71.40	73.50
ICL-TSVD	60.83	60.86	60.77	79.50	79.60	79.70	92.46	92.55	92.56	**74.21**	74.39	74.39	76.82
ADAM†	52.16	53.94	55.97	70.54	70.53	70.38	82.55	82.55	82.55	35.61	35.61	35.61	60.67
RanPAC†	58.77	**57.66**	64.59	78.10	78.46	78.86	91.48	91.86	91.86	**58.65**	72.24	72.24	74.56
ICL-TSVD†	61.78	**63.56**	66.48	80.07	80.28	80.45	92.55	92.53	92.60	**74.87**	74.89	75.13	77.93
EASE*	49.28	53.88	57.05	70.33	70.68	70.84	89.85	93.48	93.49	32.43	31.77	29.00	61.84
ICL-TSVD*	61.57	63.40	66.29	80.02	80.42	80.82	92.68	92.71	92.67	75.91	75.71	75.96	78.18

6.2 Experimental Results and Analysis

Table 1 contains the main results for $q_2 = 5, 10, 20$ on 8 different CIL datasets. First observe that L2P, DualPrompt, and CodaPrompt are unstable as their accuracy varies significantly in different datasets for different values of q_2. Second, SimpleCIL, ADAM, and EASE are unstable as their performance is largely compromised on StanfordCars. Then, RanPAC is unstable with respect to q_2 as it exhibits a large performance gap compared to ICL-TSVD for $q_2 = 5$ on ImageNet-A and StanfordCars. Finally, we see ICL-TSVD has more stable performance across datasets and for varying q_2.

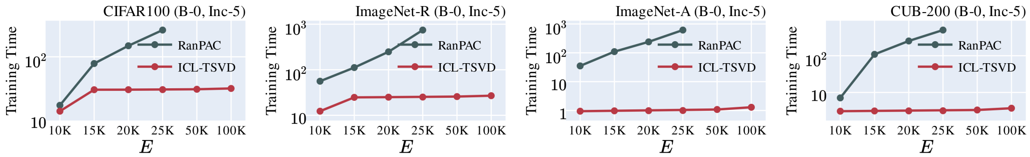


Figure 5: Training times for varying embedding dimensions E (in minutes).

Why Does ICL-TSVD Uniformly Outperform RanPAC? The first reason is that ICL-TSVD's high efficiency and scalability enable the use of a larger embedding dimension. Indeed, ICL-TSVD uses $E = 10^5$, taking advantage of the scaling law (Fig. 7, Appendix K.4), while RanPAC uses its default choice $E = 10^4$. Note that this is a fair comparison since ICL-TSVD's implementation is more scalable and more efficient than RanPAC's. Specifically, ICL-TSVD has $O(E(k_{t-1} + m_t)^2)$ time

Table 2: Final and total accuracy in CIL datasets with $q_2 = 1$ (Inc-1).

	CIFAR100	ImageNet-R	ImageNet-A	CUB	ObjectNet	OmniBenchmark	VTAB	StanfordCars	Avg.
					Final Accuracy				
RanPAC	87.06	71.78	**55.43**	**85.03**	58.51	77.89	91.11	**1.19**	**66.00**
ICL-TSVD	88.58	73.33	**62.94**	**89.36**	60.82	79.30	92.44	**74.44**	**77.65**
					Total Accuracy				
RanPAC	91.58	**70.41**	**38.25**	**73.22**	**64.94**	85.79	**77.06**	**57.43**	**69.83**
ICL-TSVD	92.88	**79.24**	**71.22**	**93.28**	**73.19**	87.01	**96.22**	**81.16**	**84.27**

complexity while RanPAC takes $O(E^3)$ time for each task t (Remark 8, Appendix C). An alternative way to make a fair comparison is to set the same embedding dimension E for both methods, in which case ICL-TSVD can be up to 1000 times faster than RanPAC (e.g., see $E = 25000$ in Fig. 5).

The second reason is that RanPAC is not stable when using a fixed regularization parameter λ (cf. Fig. 2), and the cross-validation strategy of McDonnell et al. (2023) does not fully address this challenge, as it can fail when the validation set is small. This is the case in ImageNet-A (B-0, Inc-5) and StanfordCars (B-16, Inc-5) (see Table 1), where the validation sets are small and RanPAC's performance is severly degraded. A more careful analysis of these two failure cases shows that the accuracy matrices of RanPAC have multiple columns with nearly zero entries (cf. Fig. 15a and Fig. 17c), exposing RanPAC's instability.

Inc-1: One Class at A Time. In light of the above analysis, we consider the CIL setting, with one class given at each iteration (Inc-1). In this setting, a new task has much fewer training samples and CL methods need to cope with hundreds of tasks (classes) on certain datasets. Note that adapter-based methods such as EASE are infeasible for CIL with Inc-1 (cf. Appendix H.1). In this setting, the fragility of RanPAC with respect to the choice of λ is amplified (see Table 2), and the method exhibits a significant performance drop compared to Table 1. In contrast, ICL-TSVD's performance is stable, exhibiting high accuracy comparable to the cases of Inc-$\{5, 10, 20\}$ in Table 1. Accuracy matrices associated with Table 2 are plotted in Figs. 14 to 18 of Appendix K.7.2, where we present similar results for Inc-$\{1, 2, 4, 5\}$.

7 Conclusion

This work puts forward a simple method that bridges the gap between empirical performance and theoretical guarantees in continual learning with pre-trained models. By integrating the strengths of RanPAC into the Ideal Continual Learner framework and addressing the ill-conditioning of lifted features through continual SVD truncation, our approach achieves both stability and strong performance. Extensive experiments demonstrated that our method outperforms state-of-the-art methods across multiple datasets and can handle sequences with hundreds of tasks. Theoretically, we proved that our method maintains small training and generalization errors by appropriately truncating SVD factors. This work underscores the potential of combining empirical techniques with principled frameworks to develop robust and scalable continual learning systems, and will encourage follow-up works to achieve so as well.

References

Kyra Ahrens, Hans Hergen Lehmann, Jae Hee Lee, and Stefan Wermter. Read between the layers: Leveraging multi-layer representations for rehearsal-free continual learning with pre-trained models. *Transactions on Machine Learning Research*, 2024. 1, 32, 33, 34

Anonymous Authors. The official homepage on origins of extreme learning machines (elm). https://elmorigin.wixsite.com/originofelm. Accessed: September, 2024. 34

Francis Bach. High-dimensional analysis of double descent for linear regression with random projections. *SIAM Journal on Mathematics of Data Science*, 6(1):26–50, 2024. 5, 8, 35

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in Neural Information Processing Systems*, 2019. 8, 36

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. 35

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, 2018. 35

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. 35

Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, 2002. 19

James R Bunch and Christopher P Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31(2):111–129, 1978. 19

Xufeng Cai and Jelena Diakonikolas. Last iterate convergence of incremental methods and applications in continual learning. Technical report, arXiv:2403.06873 [math.OC], 2024. 2

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. Technical report, arXiv:1902.10486v4 [cs.LG], 2019. 32

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 2022. 33

Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al. Spectral methods for data science: A statistical perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021. 32

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. 32

Meng Ding, Kaiyi Ji, Di Wang, and Jinhui Xu. Understanding forgetting in continual learning with linear regression. In *International Conference on Machine Learning*, 2024. 33

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the NTK overlap matrix. In *International Conference on Artificial Intelligence and Statistics*, 2021. 33, 48

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with dynamic token expansion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 32

Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, 2022. 2, 33

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 2020. 48

Xinyuan Gao, Songlin Dong, Yuhang He, Qiang Wang, and Yihong Gong. Beyond prompt learning: Continual adapter for efficient rehearsal-free continual learning. In *European Conference on Computer Vision*, 2024a. 33

Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. 2, 33

Daniel Goldfarb, Itay Evron, Nir Weinberger, Daniel Soudry, and PAul HAnd. The joint effect of task similarity and overparameterization on catastrophic forgetting — an analytical model. In *International Conference on Learning Representations*, 2024. 33

Alden Green and Elad Romanov. The high-dimensional asymptotics of principal component regression. Technical report, arXiv:2405.11676 [math.ST], 2024. 5, 8, 35

Etash Kumar Guha and Vihan Lakshman. On the diminishing returns of width for continual learning. In *International Conference on Machine Learning*, 2024. 39

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. 21

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of Statistics*, 50(2):949, 2022. 35

Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020. 33

Reinhard Heckel. Provable continual learning via sketched Jacobian approximations. In *International Conference on Artificial Intelligence and Statistics*, 2022. 33

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE/CVF International Conference on Computer Vision*, 2021a. 8, 36

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021b. 8, 36

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 34

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019. 33

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*, 2022. 33

Hong Hu, Yue M Lu, and Theodor Misiakiewicz. Asymptotics of random feature regression beyond the linear scaling regime. Technical report, arXiv:2403.08160 [stat.ML], 2024. 39

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *IEEE International Joint Conference on Neural Networks*, volume 2, pp. 985–990, 2004. 34

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006. 34

Ningyuan Huang, David W Hogg, and Soledad Villar. Dimensionality reduction, regularization, and generalization in overparameterized regressions. *SIAM Journal on Mathematics of Data Science*, 4 (1):126–152, 2022. 5, 8, 35

Laura Hucker and Martin Wahl. A note on the prediction error of principal component regression in high dimensions. *Theory of Probability and Mathematical Statistics*, 109:37–53, 2023. 5, 8

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *International Conference on Learning Representations*, 2021. 3

Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. Technical report, arXiv:1702.05659 [cs.LG], 2017. 3

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. 32

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, 2022. 33

Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2, 33

Doyoung Kim, Susik Yoon, Dongmin Park, Youngjun Lee, Hwanjun Song, Jihwan Bang, and Jae-Gil Lee. One size fits all for semantic shifts: Adaptive prompt tuning for continual learning. In *International Conference on Machine Learning*, 2024. 2, 33

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of The National Academy of Sciences*, 114 (13):3521–3526, 2017. 32

Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, pp. 110–133, 2017. 8

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2013. 8, 36

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009. 8, 36

Avraham Levey and Michael Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9(8):1371–1374, 2000. 19

Chuqiao Li, Zhiwu Huang, Danda Pani Paudel, Yabin Wang, Mohamad Shahbazi, Xiaopeng Hong, and Luc Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. 36, 38

Qiuwei Li, Zhihui Zhu, and Gongguo Tang. Alternating minimizations converge to second-order optimal solutions. In *International Conference on Machine Learning*, 2019. 33

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 33

Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. In *International Conference on Machine Learning*, 2023. 33

Xialei Liu. Awesome incremental learning / lifelong learning. `http://https://github.com/xialeiliu/Awesome-Incremental-Learning`, 2024. Accessed: August 2024. 32

Vincenzo Lomonaco and Davide Maltoni. CORe50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, 2017. 36, 38

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 2017. 32

Ashwani Kumar Malik, Ruobin Gao, MA Ganaie, Muhammad Tanveer, and Ponnuthurai Nagaratnam Suganthan. Random vector functional link network: recent developments, applications, and future directions. *Applied Soft Computing*, 143:110377, 2023. 34

Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 2023. 1, 3, 8, 10, 34, 36, 37, 38, 39, 48

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75 (4):667–766, 2022. 35

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013. 32

Hancheng Min, Enrique Mallada, and Rene Vidal. Early neuron alignment in two-layer ReLU networks with small initialization. In *International Conference on Learning Representations*, 2024. 2

Youngjae Min, Kwangjun Ahn, and Navid Azizan. One-pass learning via bridging orthogonal gradient descent and recursive least-squares. In *IEEE Conference on Decision and Control*, 2022. 48

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International Conference on Machine Learning*, 2022. 39

Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E. Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *IEEE/CVF International Conference on Computer Vision*, 2023. 3, 32

Y-H Pao and Yoshiyasu Takefuji. Functional-link net computing: theory, system architecture, and functionalities. *Computer*, 25(5):76–79, 1992. 3, 32, 34, 35

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 32

Binghui Peng and Andrej Risteski. Continual learning: A feature extraction formalization, an efficient algorithm, and fundamental obstructions. In *Advances in Neural Information Processing Systems*, 2022. 2, 33

Liangzu Peng, Paris Giampouras, and René Vidal. The ideal continual learner: An agent that never forgets. In *International Conference on Machine Learning*, 2023. 1, 2, 3, 19, 39, 48

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE/CVF International Conference on Computer Vision*, 2019. 36, 38

Ameya Prabhu, Shiven Sinha, Ponnurangam Kumaraguru, Philip HS Torr, Ozan Sener, and Puneet K Dokania. RanDumb: A simple approach that questions the efficacy of continual representation learning. Technical report, arXiv:2402.08823v2 [cs.CV], 2024. 1, 3, 32, 33

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 2007. 32, 35

Rahul Ramesh and Pratik Chaudhari. Model zoo: A growing brain that learns continually. In *International Conference on Learning Representations*, 2022. 33

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 32

David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77:125–141, 2008. 19

Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. Convolutional prompting meets language models for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 33

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. Technical report, arXiv:1606.04671v3 [cs.LG], 2016. 33

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. 32

Ali H Sayed. *Adaptive Filters*. John Wiley & Sons, 2008. 21

Rylan Schaeffer, Zachary Robertson, Akhilan Boopathy, Mikail Khona, Kateryna Pistunova, Jason William Rocks, Ila R Fiete, Andrey Gromov, and Sanmi Koyejo. Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle. In *Blogpost Track at ICLR 2024*, 2024. 4

Wouter F Schmidt, Martin A Kraaijveld, Robert PW Duin, et al. Feed forward neural networks with random weights. In *International Conference on Pattern Recognition*, 1992. 3, 34, 35

Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems*, 105(1):1–32, 2022. 32

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. Continual learning of large language models: A comprehensive survey. Technical report, arXiv:2404.16789 [cs.LG], 2024. 32

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 8, 33

Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. PILOT: A pre-trained model-based continual learning toolbox. Technical report, arXiv:2309.07117 [cs.LG], 2023. 8, 36, 38

William Swartworth, Deanna Needell, Rachel Ward, Mark Kong, and Halyun Jeong. Nearly optimal bounds for cyclic forgetting. *Advances in Neural Information Processing Systems*, 2023. 33

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual vitransformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 33

Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. When prompt-based incremental learning does not meet strong pretraining. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2, 33

Matus Telgarsky. Feature selection with gradient descent on two-layer networks in low-rotation regimes. Technical report, arXiv:2208.02789 [cs.LG], 2022. 2

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999. 7

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023. 35, 39

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022. 32

René Vidal, Yi Ma, and Shankar Sastry. *Generalized Principal Component Analysis*. Springer, 2016. 7

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 8, 36

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. 8

Lipo P. Wang and Chunru R. Wan. Comments on "the extreme learning machine". *IEEE Transactions on Neural Networks*, 19(8):1494–1495, 2008. 34

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 2023. 2, 33

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, 2024. 32

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 2022a. 1, 2, 33

Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022b. 2, 33

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022c. 2, 8, 33

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022d. 1, 8

Andrew R Webb and David Lowe. The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis. *Neural Networks*, 3(4):367–375, 1990. 34

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 8

Ji Xu and Daniel J Hsu. On the number of variables to use in principal component regression. *Advances in Neural Information Processing Systems*, 2019. 5, 8, 35

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically expandable representation for class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 32

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 33

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019. 32

Hongyuan Zha and Horst D Simon. On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791, 1999. 19

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. Technical report, arXiv:1910.04867v2 [cs.CV], 2019. 8, 36

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE/CVF International Conference on Computer Vision*, 2023. 33

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *European Conference on Computer Vision*, 2022. 8, 36

Xuyang Zhao, Huiyuan Wang, Weiran Huang, and Wei Lin. A statistical theory of regularization-based continual learning. In *International Conference on Machine Learning*, 2024. 33

Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. Technical report, arXiv:2303.07338 [cs.LG], 2023. 1, 2, 8, 32, 33, 36

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *International Joint Conference on Artificial Intelligence*, 2024a. 1, 3

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. 1, 8, 9, 33, 37, 38

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024c. 32

Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. *Advances in Neural Information Processing Systems*, 2022. 21

Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 21

A Overview of The Appendix

- In Appendix B, we complie all the mathematical notations used throughout the paper.
- In Appendix C we describe the implementation details of ICL-TSVD. There, we also discuss other potential implementations and our design choice.
- In Appendix D, we present auxiliary lemmas that are useful for proving our main theorems.
- In Appendix E, we prove the theorems displayed in the main paper (Theorems 1 and 2).
- In Appendix F we present results similar to Appendix E, with the difference that we now assume the noise is Gaussian, which gives slightly tighter error bounds.
- In Appendix G, we prove some extra theoretical results such as perturbation bounds on eigenvalues and eigenvectors (Theorem 6).
- In Appendix H we review related works on continual learning, focusing on CL methods with pretrained models and existing theoretical developments.
- In Appendix I we report the statistics of the datasets we use for experiments.
- In Appendix J we specify the experimental setup.
- In Appendix K we report extra experimental results, figures, and tables.

B Notations

Here in Table 3 we compile all the notations used in the paper.

Table 3: Notations

d	dimension of pre-trained features
E	embedding dimension
m_t	number of training samples for task t
M_t	$m_1 + \cdots + m_t$
c_t	total number of classes seen in the first t tasks
T	total number of tasks
$\mathcal{N}(0,1)$	Gaussian distribution with mean 0 and variance 1
$\lceil \cdot \rceil$	the ceiling operator, which maps a number to the closest integer no smaller than it
$\boldsymbol{I}_E$	$E \times E$ identity matrix
$\boldsymbol{X}_t$	$d \times m_t$ matrix, whose columns are output features of pre-trained models
$\boldsymbol{P}$	$E \times d$ random embedding matrix with $\mathcal{N}(0,1)$ entries
$\boldsymbol{H}_t$	Random ReLU features $\mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$ as defined in (1)
λ	ridge regularization parameter in RanPAC
$\widetilde{\boldsymbol{B}}_t$	the matrix whose SVDs are truncated by Algorithm 4, defined in (3)
k_t	the number of singular values and vectors preserved for the first t tasks
$\tau_{k_t}(\cdot)$	function that computes the best rank-k_t approximation of a matrix
$\mu_k(\cdot)$	the k-th largest eigenvalue of a symmetric matrix
$\boldsymbol{U}_{1:t}\boldsymbol{\Sigma}_{1:t}\boldsymbol{V}_{1:t}^\top$	SVD of $\boldsymbol{H}_{1:t}$
$\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}\overline{\boldsymbol{V}}_{1:t}^\top$	SVD of $\tau_{k_t}(\boldsymbol{H}_{1:t})$
$\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}$	SVD factors of $\widetilde{\boldsymbol{B}}_t$
a_t	accumulative error defined in (6)
γ_t	the eigengap between the present and past, defined in (5)

C Implementation Details for ICL-TSVD

In this section, we give full details of our algorithm for ICL-TSVD. Note that Algorithm 1 of the main paper is a concise version of our approach, used to illustrate the methodology at high level.

In Appendix C.1, we introduce Algorithm 2, our implementation of the incremental SVD approach. Note that Algorithm 2 dates back at least to Bunch & Nielsen (1978) and has been applied to image processing, computer vision, and latent semantic indexing (Zha & Simon, 1999; Levey & Lindenbaum, 2000; Brand, 2002; Ross et al., 2008). Recently a link between continual learning and incremental SVD was built (Peng et al., 2023). However, it has not been applied to the context we consider here to the best of our knowledge, and suitable modifications are needed to incorporate incremental SVD for solving ICL-TSVD satisfactorily. For example, we truncate the SVD factors in each continual update as shown Algorithm 2, and the outputs $(\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t})$ of Algorithm 2 are not necessarily equal to the top-k_t SVD factors of $\boldsymbol{H}_{1:t}$. It is then our contribution to arm Algorithm 2 with theoretical guarantees (cf. Lemma 1 and Theorem 6).

In Appendix C.2, we introduce Algorithm 4, a continual learning method that stably solves ICL-TSVD.

In Appendix C.3, we examine other algorithmic options that could be used for solving the least-squares type problems that we encountered in the main paper (ICL-TSVD, Min-Norm ICL, or RanPAC).

Our PyTorch code that implements ICL-TSVD will be made available soon.

C.1 Incremental Truncated SVD

We first explain the design choice as suggested by (2): Should we maintain all SVD factors $\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}$, and $\widetilde{\boldsymbol{V}}_{1:t}$, or should we just maintain the singular values $\widetilde{\boldsymbol{\Sigma}}_{1:t}$ and and left singular vectors $\widetilde{\boldsymbol{U}}_{1:t}$? In the main paper, we suggested taking the latter choice, as we empirically found continually updating all SVD factors $\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}$, and $\widetilde{\boldsymbol{V}}_{1:t}$ lead to large test errors.

We now describe how to update the top k_t SVD factors $\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}$ from the previous estimates $\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}$ and new data $\boldsymbol{H}_t$. Let $\boldsymbol{Q}_t\boldsymbol{R}_t$ be the QR decomposition of $(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{U}}_{1:t-1}^\top)\boldsymbol{H}_t$. Then we have

$$\begin{bmatrix}\widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{\Sigma}}_{1:t-1}, & \boldsymbol{H}_t\end{bmatrix} = \begin{bmatrix}\widetilde{\boldsymbol{U}}_{1:t-1}, & \boldsymbol{Q}_t\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{\Sigma}}_{1:t-1} & \widetilde{\boldsymbol{U}}_{t-1}^\top \boldsymbol{H}_t \\ 0 & \boldsymbol{R}_t\end{bmatrix}.$$

Note that $[\widetilde{\boldsymbol{U}}_{1:t-1}, \ \boldsymbol{Q}_t]$ is already orthogonal, we can do a truncated SVD on the smaller $(k_{t-1} + m_t) \times (k_{t-1} + m_t)$ matrix of the right-hand side. The full procedure is summarized below:

Algorithm 2: Incremental Truncated Singular Value Decomposition

```
Input: data matrix $\boldsymbol{H}_t \in \mathbb{R}^{E\times m_t}$ of task $t$, desired output rank $k_t \leq m$, SVD factors
    $\widetilde{\boldsymbol{U}}_{1:t-1} \in \mathbb{R}^{E\times k_{t-1}}$ and $\widetilde{\boldsymbol{\Sigma}}_{1:t-1} \in \mathbb{R}^{k_{t-1}\times k_{t-1}}$ of previous $t-1$ tasks;
Compute the QR decomposition $\boldsymbol{Q}_t\boldsymbol{R}_t$ of $(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{U}}_{1:t-1}^\top)\boldsymbol{H}_t$;
Set $(\boldsymbol{\Sigma}_{\text{tmp}}, \boldsymbol{U}_{\text{tmp}})$ to the top-$k_t$ SVD components of          // TSVD (Algorithm 3)

    $\begin{bmatrix}\widetilde{\boldsymbol{\Sigma}}_{1:t-1} & \widetilde{\boldsymbol{U}}_{t-1}^\top \boldsymbol{H}_t \\ 0 & \boldsymbol{R}_t\end{bmatrix} \in \mathbb{R}^{(k_{t-1}+m_t)\times(k_{t-1}+m_t)};$          (11)

Set $\widetilde{\boldsymbol{\Sigma}}_{1:t} \leftarrow \boldsymbol{\Sigma}_{\text{tmp}}$ and $\widetilde{\boldsymbol{U}}_{1:t} \leftarrow [\widetilde{\boldsymbol{U}}_{t-1} \ \boldsymbol{Q}_t]\boldsymbol{U}_{\text{tmp}}$;
$\widetilde{\boldsymbol{U}}_{1:t} \leftarrow$ The orthogonal factor of QR decomposition of $\widetilde{\boldsymbol{U}}_{1:t}$;          // improve numerical stability
Output: $(\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t})$;
```

Remark 7. Since $[\widetilde{\boldsymbol{U}}_{1:t-1} \ \boldsymbol{Q}_t]$ and $\boldsymbol{U}_{\text{tmp}}$ are orthogonal, $\widetilde{\boldsymbol{U}}_{1:t}$ is expected to be orthogonal as well. However, the multiplication $\widetilde{\boldsymbol{U}}_{1:t} = [\widetilde{\boldsymbol{U}}_{t-1} \ \boldsymbol{Q}_t]\boldsymbol{U}_{\text{tmp}}$ might lose orthogonality due to numerical errors, especially for large t. This is fixed by an extra post-processing step that orthogonalizes $\widetilde{\boldsymbol{U}}_{1:t}$.

Algorithm 3: Truncated Singular Value Decomposition (TSVD)

```
Input: matrix $\boldsymbol{H} \in \mathbb{R}^{E \times m}$ and desired output rank $r \leq \min\{E, m\}$;
tmp $\leftarrow \min\{E, m\}$;
Compute the SVD $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \cdots + \sigma_{\text{tmp}} \boldsymbol{u}_{\text{tmp}} \boldsymbol{v}_{\text{tmp}}^\top$ of $\boldsymbol{H}$;
Set $\widetilde{\boldsymbol{\Sigma}} \leftarrow \operatorname{diag}(\sigma_1, \dots, \sigma_r), \widetilde{\boldsymbol{U}} \leftarrow [\boldsymbol{u}_1, \dots, \boldsymbol{u}_r]$;
Output: $(\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{\Sigma}})$;
```

Memory Complexity Analysis. The extra working memory of this approach is roughly:

- $O(Em_t + m_t^2)$, for the QR factors $\boldsymbol{Q}_t \boldsymbol{R}_t$;
- $O((k_{t-1} + m_t)^2)$, for the matrix in (11) and its SVD factors;

Hence, for large E, this is less than the $O(E(k_{t-1} + m_t))$ memory used by the direct SVD method.

Time Complexity Analysis. The major cost is the SVD of (11), which takes $O((k_{t-1} + m_t)^3)$ time. While in principle the QR orthogonalization for the post-processing of $\widetilde{\boldsymbol{U}}_{1:t}$ takes $O(E(k_{t-1} + m_t)^2)$ time, it is significantly faster than SVD as the constants behind its $O(\cdot)$ is very small. Therefore, one would expect the SVD on the matrix of (11) in $O((k_{t-1} + m_t)^3)$ time should be much faster than the SVD on the matrix $[\widetilde{\boldsymbol{U}}_{1:t-1} \widetilde{\boldsymbol{\Sigma}}_{1:t-1},\ \boldsymbol{H}_t]$, which needs $O(E(k_{t-1} + m_t)^2)$ time, where E is far larger than $k_{t-1} + m_t$ (e.g., $E = 10^5$ and $k_{t-1} + m_t = 10^4$). This is true on a sequential machine, but their running time difference is not significant for highly parallel GPU implementations in our experience (e.g., computing the inner product between two E-dimensional vectors has similar running times to computing the inner product between $(k_{t-1} + m_t)$-dimensional vectors, due to parallelism). Hence, for a parallel implementation, the main advantage of doing SVD on the matrix in (11) is that it takes less working memory than SVD on $[\widetilde{\boldsymbol{U}}_{1:t-1} \widetilde{\boldsymbol{\Sigma}}_{1:t-1},\ \boldsymbol{H}_t]$.

Remark 8. While our method has $O(E(k_{t-1} + m_t)^2)$ time complexity, RanPAC solves the normal equations $\boldsymbol{W}(\boldsymbol{H}_{1:t} \boldsymbol{H}_{1:t}^\top + \lambda \boldsymbol{I}_E) = \boldsymbol{Y}_{1:t} \boldsymbol{H}_{1:t}^\top$ of the ridge problem in variable $\boldsymbol{W}$ for every task t using off-the-shelf solvers implemented in PyTorch, which in general takes $O(E^3)$ time. This is why it is slower than our method for the same E, particularly when E is large (Figs. 5 and 6).

C.2 Continual Solver for ICL-TSVD

The proposed algorithm is shown in Algorithm 4. Here are a few details that we have not yet mentioned in the main paper. First, note that Algorithm 4 formally updates M_t ad $\boldsymbol{J}_{1:t}$ continually. At Line 8 of Algorithm 4 we compute the label-feature covariance matrix $\boldsymbol{J}_1 := \boldsymbol{Y}_1 \boldsymbol{H}_1^\top \in \mathbb{R}^{c_1 \times E}$, and then at lines 10 and 11 we update $\boldsymbol{J}_{1:t-1}$ into $\boldsymbol{J}_{1:t}$ via $\boldsymbol{J}_{1:t} \leftarrow \boldsymbol{J}_{1:t-1} + \boldsymbol{J}_{\text{tmp}}$. The attentive reader might find that $\boldsymbol{J}_{1:t-1}$ is of size $c_{t-1} \times E$ while $\boldsymbol{J}_{\text{tmp}}$ is of size $c_t \times E$. But it could be that $c_{t-1} < c_t$, so it might not make sense to add $\boldsymbol{J}_{1:t-1}$ and $\boldsymbol{J}_{\text{tmp}}$ as in Line 11. Note that we wrote Line 11 just for simplicity. The implementation would pad $c_t - c_{t-1}$ zero rows to $\boldsymbol{J}_{1:t-1}$ in a similar fashion to how we extend $\boldsymbol{Y}_{t-1}$ into $\boldsymbol{Y}_t$ when more classes are given, and this is what Line 11 should mean.

Second, we add an extra parameter r_{max}, to control the *maximum allowable rank*, that is the maximum number of columns $\widetilde{\boldsymbol{U}}_{1:t}$ is allowed to have. The purpose is to control the time complexity of Algorithm 4 and allow it to run more efficiently on large datasets such as DomainNet (cf. Tables 7 and 8). We argue both the truncation percentage ζ and maximum allowable rank r_{max} are needed: With ζ alone, the method might run slowly or even exceed the memory for large datasets such as DomainNet (cf. Tables 7 and 8); with r_{max} alone, truncation is not activated before receiving r_{max} samples, and numerical instability if it arises, can not be prevented before truncation is in effect. Table 4 gives the values of ζ and r_{max} we use for each dataset.

C.3 Alternative approaches for online Least-Squares or Ridge Regression

Here we examine a few alternative options of solving the least-squares or ridge regression problems that appeared in the main paper (namely, Min-Norm ICL, ICL-TSVD, RanPAC).

Dataset	Truncation Percentage ζ	Embedding Dimension E	Maximum Allowable Rank r_{max}
CIFAR100	25%	10^5	10000
ImageNet-R	25%	10^5	10000
ImageNet-A	25%	10^5	10000
CUB-200	25%	10^5	10000
ObjectNet	25%	10^5	20000
OmniBenchmark	25%	10^5	20000
VTAB†	25%	10^5	10000
StanfordCars	25%	10^5	10000

Table 4: Hyperparameters we use for each dataset.

Algorithm 4: Continual Solver of ICL-TSVD (concise version in Algorithm 1)

```
Input of Task t: Random ReLU features $\boldsymbol{H}_t \in \mathbb{R}^{E\times m_t}$, label matrix $\boldsymbol{Y}_t \in \mathbb{R}^{c_t\times m_t}$,
  truncation percentage $\zeta \in [0,1]$, maximum allowable rank $r_{\text{max}}$;
For $t \leftarrow 1, 2, \dots$:
    $M_t \leftarrow M_{t-1} + m_t$;                                  // update the total number of samples $M_t$
    $k_t \leftarrow \min\{r_{\text{max}}, (1-\zeta)M_t)\}$;          // perserve $k_t$ SVD factors for the first $t$ tasks
    Form $\widetilde{\boldsymbol{B}}_t$ as per (3);
    $(\widetilde{\boldsymbol{U}}_{1:t}, \widetilde{\boldsymbol{\Sigma}}_{1:t}) \leftarrow$ Top-$k_t$ SVD factors of $\widetilde{\boldsymbol{B}}_t$;   // use Algorithm 3 if $t=1$, or Algorithm 2 if $t>1$
    If $t = 1$:                                                      // Continual update of $\boldsymbol{J}_{1:t} := \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$
        $\boldsymbol{J}_1 \leftarrow$ Output of Algorithm 5 run with inputs $\boldsymbol{H}_1, \boldsymbol{Y}_1$;   // label-feature covariance of task 1
    Else:
        $\boldsymbol{J}_{\text{tmp}} \leftarrow$ Output of Algorithm 5 run with inputs $\boldsymbol{H}_t, \boldsymbol{Y}_t$;   // label-feature covariance of task $t$
        $\boldsymbol{J}_{1:t} \leftarrow \boldsymbol{J}_{1:t-1} + \boldsymbol{J}_{\text{tmp}}$;
    Form the linear classifier $\widetilde{\boldsymbol{W}}_t := \boldsymbol{J}_{1:t}\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)$;   // cf. (2) and (4)
```

Randomized Linear Algebra Techniques. Algorithm 3 can, in fact, be implemented by randomized linear algebra techniques (Halko et al., 2011). Some of these techniques compute by design only the top k SVD factors. Intuitively this could save time and memory if k is extremely small. One such method is conveniently implemented in PyTorch as well (`torch.svd_lowrank`). However, in our rudimentary attempts at using randomized approaches, we found this PyTorch routine does not seem to be as efficient or accurate as our present implementation (we consistently set truncation percentage ζ to 25%). This observation aligns with the PyTorch document of `torch.svd_lowrank`: *In general, use the full-rank SVD implementation torch.linalg.svd() for dense matrices due to its 10-fold higher performance characteristics. The low-rank SVD will be useful for huge sparse matrices that torch.linalg.svd() cannot handle*. For the moment, we conclude that it needs deeper investigations to see whether randomized techniques are suitable for the continual learning contexts.

Online Matrix Inversion. A key step in solving Min-Norm ICL or RanPAC would be computing the inverse $(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \lambda \boldsymbol{I}_E)^{-1}$ or $(\boldsymbol{H}_{1:t}^\top\boldsymbol{H}_{1:t} + \lambda \boldsymbol{I}_{M_t})^{-1}$. Their inversion can be continually updated via the celebrated Woodbury matrix identity or block matrix inversion lemma, and the signature use of these matrix tools can be found in classic signal processing algorithms such as *recursive least-squares* Sayed (2008). They are also used in recent continual learning papers of Zhuang et al. (2022; 2023). Using them allows the inversion to be computed rapidly, while it is also known that they can be numerically unstable. Indeed, in our setting, $\boldsymbol{H}_{1:t}$ is extremely ill-conditioned. Our implementation where $(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \lambda \boldsymbol{I}_E)^{-1}$ are continually updated via the Woodbury matrix identity suffers from numerical errors, and it is unable to maintain strong performance and unable to outperform RanPAC. Moreover, the numerical errors accumulate over time and can not handle long sequences of tasks. Finally, even if we know the numerical errors exist, there is no obvious solution to fix them. This is different from our ICL-TSVD implementation based on robust truncated SVD procedures, which has an advantage that we could easily reduce numerical errors by re-orthogonalizing $\widetilde{\boldsymbol{U}}_{1:t}$ (see Remark 7 and Algorithm 2).

Algorithm 5: Compute The Label-Feature Covariance Matrix

```
Input: matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \dots, \boldsymbol{h}_m] \in \mathbb{R}^{E \times m}$ and label matrix $\boldsymbol{Y} \in \mathbb{R}^{c \times m}$;
Convert $\boldsymbol{Y}$ into a vector of indices $\boldsymbol{y} = [y_1, \dots, y_m]$ such that the $i$-th column of $\boldsymbol{Y}$ is the $y_i$-th
  standard basis vector (i.e., one-hot vector with 1 at position $y_i$);
Initialize $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_c]$ to be the $E \times c$ zero matrix;
For $i = 1, \dots, m$:                     // parallel implementation via torch.Tensor.index_add_
    $\boldsymbol{S}_{y_i} \leftarrow \boldsymbol{S}_{y_i} + \boldsymbol{h}_i$;
Output: $\boldsymbol{S}^\top$;                                              // $\boldsymbol{S}$ is equal to $\boldsymbol{H}\boldsymbol{Y}^\top$
```

D Auxiliary Lemmas

The following lemma provides an explicit expression for the difference between the continually truncated factors $\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top$ and covariance $\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top$.

Lemma 1. *Let $\widetilde{\boldsymbol{B}}_t$ be the matrix whose SVDs are truncated by Algorithm 4, as defined in (3). We have $\overline{\boldsymbol{U}}_1 = \widetilde{\boldsymbol{U}}_1$ and $\overline{\boldsymbol{\Sigma}}_1 = \widetilde{\boldsymbol{\Sigma}}_1$. Moreover, we have*

$$\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top = \sum_{i=1}^{t}\left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big)\right).$$

Proof of Lemma 1. It should be clear that $\overline{\boldsymbol{U}}_1 = \widetilde{\boldsymbol{U}}_1$ and $\overline{\boldsymbol{\Sigma}}_1 = \widetilde{\boldsymbol{\Sigma}}_1$. For every $i = 1, \dots, t$ we have

$$\widetilde{\boldsymbol{U}}_{1:i}\widetilde{\boldsymbol{\Sigma}}_{1:i}^2\widetilde{\boldsymbol{U}}_{1:i}^\top = \tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big).$$

and therefore

$$\widetilde{\boldsymbol{U}}_{1:i}\widetilde{\boldsymbol{\Sigma}}_{1:i}^2\widetilde{\boldsymbol{U}}_{1:i}^\top - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top = \tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top.$$

A key observation is that summing the above equality over $i = 2, \dots, t$ yields

$$\begin{aligned}
&\sum_{i=2}^{t}\left(\widetilde{\boldsymbol{U}}_{1:i}\widetilde{\boldsymbol{\Sigma}}_{1:i}^2\widetilde{\boldsymbol{U}}_{1:i}^\top - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right) = \sum_{i=2}^{t}\left(\tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right) \\
\Leftrightarrow &\sum_{i=2}^{t}\left(\widetilde{\boldsymbol{U}}_{1:i}\widetilde{\boldsymbol{\Sigma}}_{1:i}^2\widetilde{\boldsymbol{U}}_{1:i}^\top - \widetilde{\boldsymbol{U}}_{1:i-1}\widetilde{\boldsymbol{\Sigma}}_{1:i-1}^2\widetilde{\boldsymbol{U}}_{1:i-1}^\top - \boldsymbol{H}_i\boldsymbol{H}_i^\top\right) = \sum_{i=2}^{t}\left(\tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right) \\
\Leftrightarrow &\,\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_1\widetilde{\boldsymbol{\Sigma}}_1^2\widetilde{\boldsymbol{U}}_1^\top - \boldsymbol{H}_{2:t}\boldsymbol{H}_{2:t}^\top = \sum_{i=2}^{t}\left(\tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right) \qquad (12) \\
\Leftrightarrow &\,\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top = \overline{\boldsymbol{U}}_1\overline{\boldsymbol{\Sigma}}_1^2\overline{\boldsymbol{U}}_1^\top - \boldsymbol{H}_1\boldsymbol{H}_1^\top + \sum_{i=2}^{t}\left(\tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right) \\
\Leftrightarrow &\,\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top = \sum_{i=1}^{t}\left(\tau_{k_i}\big(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\big) - \widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right).
\end{aligned}$$

The last equality also holds for $t = 1$. This finishes the proof. □

The lemma below is a direct consequence of Von Neumann's trace inequality, and its proof is omitted.

Lemma 2. *Given two square matrices A, B with A positive semidefinite, we have*

$$\operatorname{tr}(AB) \le \operatorname{tr}(A) \cdot \|B\|.$$

Lemma 3 presented below is elementary.

Lemma 3. *Assume C is a positive semidefinite matrix. Then we have*

$$\operatorname{tr}(DACBD^\top) + \operatorname{tr}(DB^\top CA^\top D^\top) \le \operatorname{tr}(DACA^\top D^\top) + \operatorname{tr}(DBCB^\top D^\top),$$

where A, B, C, D are matrices of compatible sizes. Therefore, it holds that

$$\operatorname{tr}\big(D(A+B)C(A+B)D^\top\big) \le 2\operatorname{tr}(DACA^\top D^\top) + 2\operatorname{tr}(DBCB^\top D^\top).$$

The following two lemmas provide upper bounds on several terms appearing naturally in our main results.

Lemma 4. *Let $\widetilde{\boldsymbol{B}}_t$ be defined in (3), γ_t in (5), and a_t in (6). Define*

$$\boldsymbol{D}_t := \sum_{i=1}^{t} \left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}\left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\right)\right). \tag{13}$$

We have

$$\begin{aligned}
\left\|\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right\| &\leq \frac{t-1}{\gamma_t}, \\
\left\|\widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{D}_t\right\| &\leq \frac{t-1}{\gamma_t}, \\
\operatorname{tr}\left(\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) &\leq \frac{1}{\gamma_t} \min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\}, \\
\operatorname{tr}\left(\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-4} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) &\leq \frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\right)} \cdot \frac{1}{\gamma_t} \min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\}.
\end{aligned}$$

Proof of Lemma 4. It follows from definition that

$$\left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}\left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\right)\right) \widetilde{\boldsymbol{U}}_{1:t} = 0,$$

hence $\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} = \boldsymbol{D}_{t-1} \widetilde{\boldsymbol{U}}_{1:t}$. This means

$$\begin{aligned}
\left\|\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right\| &= \left\|\boldsymbol{D}_{t-1} \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right\| \\
&\leq \|\boldsymbol{D}_{t-1}\| \cdot \left\|\widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right\| \\
&= \|\boldsymbol{D}_{t-1}\| \cdot \frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\right)},
\end{aligned}$$

where the last equality follows by definition. On the other hand, we have

$$\|\boldsymbol{D}_{t-1}\| \leq \sum_{i=1}^{t-1} \mu_{k_i+1}\left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\right) \leq \frac{(t-1)}{\gamma_t} \mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\right).$$

Combining the above proves the first required equality. The second inequality follows similarly.

For the final trace inequality, we have ($k_0 := 0$)

$$\begin{aligned}
\operatorname{tr}\left(\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) &= \operatorname{tr}\left(\boldsymbol{D}_{t-1} \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) \\
&\overset{\text{(i)}}{\leq} \operatorname{tr}(\boldsymbol{D}_{t-1}) \cdot \left\|\widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right\| \\
&= \left(\sum_{i=1}^{t-1} \sum_{j=k_i+1}^{m_i+k_{i-1}} \mu_j\left(\widetilde{\boldsymbol{B}}_i \widetilde{\boldsymbol{B}}_i^\top\right)\right) \cdot \frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top\right)} \\
&\leq \frac{1}{\gamma_t} \sum_{i=1}^{t-1} (m_i + k_{i-1} - k_i) \\
&= \frac{1}{\gamma_t} (M_{t-1} - k_{t-1}),
\end{aligned}$$

where (i) holds as $\boldsymbol{D}_{t-1}$ is positive semidefinite (cf. Lemma 2).

We can also bound $\operatorname{tr}\left(\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right)$ alternatively as follows:

$$\begin{aligned}
\operatorname{tr}\left(\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) &= \operatorname{tr}\left(\boldsymbol{D}_{t-1} \widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top\right) \\
&\leq \|\boldsymbol{D}_{t-1}\| \cdot \operatorname{tr}(\widetilde{\boldsymbol{U}}_{1:t} \widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2} \widetilde{\boldsymbol{U}}_{1:t}^\top) \\
&\leq \frac{(t-1)k_t}{\gamma_t}
\end{aligned}$$

Combining the two bounds on $\operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)$ proves the third inequality. The fourth inequality follows similarly. □

Lemma 5. *Using the notations in Lemma 4, we have*

$$\operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) \le \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\},$$

$$\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\| \le a_{t-1}\cdot\left(\frac{(t-1)^2}{\gamma_t^2}+\frac{t-1}{\gamma_t}\right),$$

$$\left\|(\boldsymbol{I}_E-\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\boldsymbol{I}_E-\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\| \le a_t,$$

$$\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \le \left(\frac{t-1}{\gamma_t^2}+\frac{2}{\gamma_t}\right)\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}+k_t,$$

$$\operatorname{tr}\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) \le \frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top\right)}\cdot\left(\frac{\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}}{\gamma_t}+k_t\right).$$

Proof of Lemma 5. Since $\boldsymbol{D}_{t-1}$ is positive semidefinite, let $\boldsymbol{L}_{t-1}\boldsymbol{L}_{t-1}^\top$ be its Cholesky decomposition. Then we have

$$\begin{aligned}
&\operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
=\ &\operatorname{tr}\left(\boldsymbol{D}_{t-1}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_{t-1}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
=\ &\operatorname{tr}\left(\boldsymbol{L}_{t-1}\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
=\ &\operatorname{tr}\left(\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right)\\
\overset{\text{(i)}}{\le}\ &\operatorname{tr}\left(\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right)\cdot\left\|\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right\|\\
=\ &\operatorname{tr}\left(\boldsymbol{D}_{t-1}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\cdot\left\|\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right\|\\
\overset{\text{(ii)}}{\le}\ &\frac{1}{\gamma_t}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}\cdot\left\|\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right\|.
\end{aligned}$$

In the above, (i) follows from Lemma 2, and (ii) follows from Lemma 4. Continuing with the above inequality, we have

$$\begin{aligned}
\left\|\boldsymbol{L}_{t-1}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{L}_{t-1}\right\| &\le \left\|\boldsymbol{L}_{t-1}\right\|^2\cdot\frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top\right)}\\
&= \left\|\boldsymbol{D}_{t-1}\right\|\cdot\frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top\right)}\\
&\le \frac{t-1}{\gamma_t}.
\end{aligned}$$

Recall the fact $\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}=\boldsymbol{D}_{t-1}\widetilde{\boldsymbol{U}}_{1:t}$. The second inequality in Lemma 5 can be proved as follows:

$$\begin{aligned}
&\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|\\
\overset{\text{(i)}}{=}\ &\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top(\boldsymbol{D}_t+\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{2}\widetilde{\boldsymbol{U}}_{1:t}^\top)\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|\\
=\ &\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t+\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|\\
\le\ &\left\|\boldsymbol{D}_{t-1}\right\|\cdot\left(\left\|\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|+\left\|\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|\right)\\
\le\ &a_{t-1}\cdot\left(\frac{(t-1)^2}{\gamma_t^2}+\frac{t-1}{\gamma_t}\right).
\end{aligned}$$

In the above, (i) follows from Lemma 1.

The third inequality is proved as follows:

$$\begin{aligned}
&\left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\| \\
={}& \left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}^2\widetilde{\boldsymbol{U}}_{1:t}^\top)(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\| \\
={}& \left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{D}_t(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\| \\
\leq{}& \|\boldsymbol{D}_t\| = a_t.
\end{aligned}$$

We now prove the fourth inequality:

$$\begin{aligned}
&\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \\
={}& \operatorname{tr}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) \\
\overset{\text{(i)}}{=}{}& \operatorname{tr}\left((\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right) \\
={}& \operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + 2\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) + k_t \\
\overset{\text{(ii)}}{\leq}{}& \left(\frac{t-1}{\gamma_t^2} + \frac{2}{\gamma_t}\right)\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\} + k_t.
\end{aligned}$$

In the above, (i) follows from Lemma 1, and (ii) follows from Lemma 4 and the first inequality we just proved for Lemma 5.

The fifth inequality can be proved as follows:

$$\begin{aligned}
&\operatorname{tr}\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) \\
\overset{\text{(i)}}{=}{}& \operatorname{tr}\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right) \\
={}& \operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-4}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) + \operatorname{tr}\left(\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\right) \\
\overset{\text{(ii)}}{\leq}{}& \frac{1}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top\right)} \cdot \frac{1}{\gamma_t}\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\} + \frac{k_t}{\mu_{k_t}\left(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top\right)}.
\end{aligned}$$

Here, (i) holds as a result of Lemma 1 and (ii) follows from Lemma 4.

□

Lemma 6. *Using the notations in Lemma 4, we have for any $\boldsymbol{W}$ that*

$$\left\|\boldsymbol{W}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \leq 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2\left(a_t + \frac{a_{t-1}(t-1)}{\gamma_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2}\right).$$

Proof. We have

$$\begin{aligned}
&\left\|\boldsymbol{W}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \\
={}& \operatorname{tr}\left(\boldsymbol{W}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \boldsymbol{I}_{M_t})(\boldsymbol{W})^\top\right) \\
\overset{\text{(i)}}{\leq}{}& \operatorname{tr}\left(\boldsymbol{W}(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)(\boldsymbol{W})^\top\right) \\
\overset{\text{(ii)}}{\leq}{}& 2\operatorname{tr}\left(\boldsymbol{W}(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)(\boldsymbol{W})^\top\right) \\
&\quad + 2\operatorname{tr}\left(\boldsymbol{W}\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t(\boldsymbol{W})^\top\right) \\
\overset{\text{(iii)}}{\leq}{}& 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2 \cdot \left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\| \\
&\quad + 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2 \cdot \left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\| \\
\overset{\text{(iv)}}{\leq}{}& 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2 \cdot a_t + 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2 \cdot a_{t-1} \cdot \left(\frac{(t-1)^2}{\gamma_t^2} + \frac{t-1}{\gamma_t}\right) \\
={}& 2 \cdot \|\boldsymbol{W}\|_{\mathrm{F}}^2\left(a_t + \frac{a_{t-1}(t-1)}{\gamma_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2}\right).
\end{aligned}$$

In the above, (i) follows from Lemma 1, (ii) from Lemma 3, (iii) from Lemma 2, and (iv) from Lemma 5. The proof is complete. □

E Proofs of Theorem 1 and Theorem 2

Proof of Theorem 1. Let $\boldsymbol{I}_{M_t}$ be the $M_t \times M_t$ identity matrix. The training loss can be written as

$$\begin{aligned}
&\left\|\widetilde{\boldsymbol{W}}_t \boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \\
&= \left\|\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \\
&= \left\|\boldsymbol{Y}_{1:t}(\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
&= \left\|(\boldsymbol{W}_t^* \boldsymbol{H}_{1:t} + \mathcal{E}_{1:t})(\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
&\leq 2\cdot\left\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 + 2\cdot\left\|\mathcal{E}_{1:t}(\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2.
\end{aligned}$$

We can now bound the first term by Lemma 6 as follows:

$$\left\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \leq 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\left(a_t + \frac{a_{t-1}(t-1)}{\gamma_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2}\right).$$

The second term is bounded above as follows:

$$\begin{aligned}
&2\cdot\left\|\mathcal{E}_{1:t}(\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
\leq\ & 2\cdot\left\|\mathcal{E}_{1:t}\right\|^2\cdot\left\|(\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top \boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
\leq\ & 2\cdot\left\|\mathcal{E}_{1:t}\right\|^2\cdot\left(M_t - k_t + \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}\right),
\end{aligned}$$

where the first inequality follows from Lemma 2 and the last inequality from Proposition 1. □

Proof of Theorem 2. Define $\boldsymbol{D}_t := \sum_{i=1}^t \left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}\left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right)\right)$. Note that $\boldsymbol{D}_t$ is a symmetric and positive semi-definite matrix.

Note that we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}}\left[\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h} - \boldsymbol{y}\right\|^2\right] &= \mathbb{E}_{\boldsymbol{h}}\left[\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h} - \boldsymbol{\epsilon}\right\|^2\right] \\
&= 2\cdot\mathbb{E}_{\boldsymbol{h}}\left[\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\right\|^2\right] + 2\cdot\|\boldsymbol{\epsilon}\|^2,
\end{aligned}$$

so we next focus on bounding $\mathbb{E}_{\boldsymbol{h}}\left[\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\right\|^2\right]$. With the $E \times E$ identity matrix $\boldsymbol{I}_E$ and $\boldsymbol{\Lambda} := \mathbb{E}[\boldsymbol{h}\boldsymbol{h}^\top]$, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}}\left[\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{h}}\left[\|(\boldsymbol{W}_t^*\boldsymbol{H}_{1:t} + \mathcal{E}_{1:t})\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{h}}\left[\|(\boldsymbol{W}_t^*\boldsymbol{H}_{1:t} + \mathcal{E}_{1:t})\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\
&\leq 2\cdot\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h} - \boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] + 2\cdot\mathbb{E}_{\boldsymbol{h}}\left[\|\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2\right] \\
&\leq 2\cdot\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{h}\|^2\right] + 2\cdot\|\mathcal{E}_{1:t}\|^2\cdot\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2\right]
\end{aligned}$$

The term $\mathbb{E}_{\boldsymbol{h}}\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2$ can be bounded above as follows:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}}\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2\\
&= \operatorname{tr}\left(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{\Lambda}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right)\\
&= \operatorname{tr}\left(\left(\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right)\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
&\overset{\text{(i)}}{\leq} \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| \cdot \operatorname{tr}\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
&\quad + \frac{1}{M_t}\operatorname{tr}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\\
&\overset{\text{(ii)}}{\leq} \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| \cdot \frac{\left(\frac{1}{\gamma_t}\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\} + k_t\right)}{\mu_{k_t}(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top)}\\
&\quad + \frac{1}{M_t\gamma_t}\left(\frac{t-1}{\gamma_t^2 M_t} + \frac{2}{\gamma_t M_t}\right)\cdot\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\} + \frac{k_t}{M_t}\\
&=: \mathbb{V}_t
\end{aligned}
$$

Here, (i) follows from Lemma 2 and (ii) follows from Lemma 5.

The term $\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{h}\|^2\right]$ satisfies:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{h}\|^2\right]\\
&= \operatorname{tr}\left(\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{\Lambda}(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \boldsymbol{I}_E)(\boldsymbol{W}_t^*)^\top\right)\\
&\overset{\text{(i)}}{=} \operatorname{tr}\left(\boldsymbol{W}_t^*(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{\Lambda}(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t + \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)(\boldsymbol{W}_t^*)^\top\right)\\
&\overset{\text{(ii)}}{\leq} 2\operatorname{tr}\left(\boldsymbol{W}_t^*(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{\Lambda}(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)(\boldsymbol{W}_t^*)^\top\right)\\
&\quad + 2\operatorname{tr}\left(\boldsymbol{W}_t^*\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{\Lambda}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t(\boldsymbol{W}_t^*)^\top\right)\\
&\overset{\text{(iii)}}{\leq} 2\cdot\|\boldsymbol{W}_t^*(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\|_{\mathrm{F}}^2\cdot\left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{\Lambda}(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\|\\
&\quad + 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\cdot\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{\Lambda}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|
\end{aligned}
$$

where the above three steps, (i), (ii), and (iii), follow from Lemma 1, Lemma 3, and Lemma 2 respectively. To bound $\mathbb{B}_{t1} := \left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\boldsymbol{\Lambda}(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\|$, we have

$$
\begin{aligned}
\mathbb{B}_{t1} &= \left\|(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\left(\boldsymbol{\Lambda} - \frac{1}{M_t}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top\right)(\boldsymbol{I}_E - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top)\right\|\\
&\leq \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top\right\|\\
&\leq \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| + \frac{1}{M_t}\cdot\left\|\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^2\widetilde{\boldsymbol{U}}_{1:t}^\top\right\|\\
&= \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| + \frac{1}{M_t}\cdot\|\boldsymbol{D}_t\|\\
&= \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| + \frac{a_t}{M_t}.
\end{aligned}
$$

To bound $\mathbb{B}_{t2} := \left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{\Lambda}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|$, we have

$$\begin{aligned}
\mathbb{B}_{t2} &= \left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{\Lambda}\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\| \\
&= \left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\left(\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right)\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\| \\
&\leq \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| \cdot \left\|\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\|^2 \\
&\quad + \frac{1}{M_t}\left\|\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\right\| \\
&\leq \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| \cdot \frac{(t-1)^2}{\gamma_t^2} + a_{t-1}\cdot\left(\frac{(t-1)^2}{\gamma_t^2} + \frac{t-1}{\gamma_t}\right),
\end{aligned}$$

where the last step follows from Lemma 5. Putting together, we have obtained

$$\begin{aligned}
&\frac{\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_E)\boldsymbol{h}\|^2\right]}{2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2} \\
&\leq \mathbb{B}_{t1} + \mathbb{B}_{t2} \\
&\leq \left\|\boldsymbol{\Lambda} - \frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\| \cdot \left(1 + \frac{(t-1)^2}{\gamma_t^2}\right) + \frac{a_{t-1}}{M_t}\cdot\left(\frac{(t-1)^2}{\gamma_t^2} + \frac{t-1}{\gamma_t}\right) + \frac{a_t}{M_t} =: \mathbb{B}_t.
\end{aligned}$$

Combining the above finishes the proof. □

F Theoretical Guarantees under Gaussian Assumptions

In this section, we prove slightly tighter results than Theorems 1 and 2 presented in the main paper. The key idea is to make certain Gaussian assumptions on noise. Specifically, we assume both the training noise $\mathcal{E}_{1:t}$ and test noise $\boldsymbol{\epsilon}$ have i.i.d. $\mathcal{N}(0,\nu^2)$ entries. With these, we present and prove Theorems 3 to 5 below.

Theorem 3. *On top of the settings of Theorem 1, furthermore assume $\mathcal{E}_{1:t}$ consists of i.i.d. $\mathcal{N}(0,\nu^2)$ entries. Then the output $\widetilde{\boldsymbol{W}}_t = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$ of our method (4) satisfies*

$$\begin{aligned}
\frac{1}{M_t}\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \leq 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\left(\frac{a_t}{M_t} + \frac{a_{t-1}(t-1)}{\gamma_t M_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2 M_t}\right) \qquad (14) \\
+ c_t\nu^2\left(\frac{(M_t-k_t)}{M_t} + \frac{(t-1)\min\{M_{t-1}-k_{t-1},(t-1)k_t\}}{\gamma_t^2 M_t}\right).
\end{aligned}$$

Proof of Theorem 3. Let $\boldsymbol{I}_{M_t}$ be the $M_t\times M_t$ identity matrix. The training loss can be written as

$$\begin{aligned}
&\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_{\mathcal{E}_{1:t}}\left\|\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_{\mathcal{E}_{1:t}}\left\|\boldsymbol{Y}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_{\mathcal{E}_{1:t}}\left\|(\boldsymbol{W}_t^*\boldsymbol{H}_{1:t} + \mathcal{E}_{1:t})(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2 \\
&= \left\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 + \mathbb{E}_{\mathcal{E}_{1:t}}\left\|\mathcal{E}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2.
\end{aligned}$$

We can now bound the first term by Lemma 6 as follows:

$$\left\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2 \leq 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\left(a_t + \frac{a_{t-1}(t-1)}{\gamma_t} + \frac{a_{t-1}(t-1)^2}{\gamma_t^2}\right).$$

The second term can be bounded above as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{E}}\left\|\mathcal{E}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2\\
=\,&\mathbb{E}_{\mathcal{E}}\operatorname{tr}\left((\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})\mathcal{E}_{1:t}{}^\top\mathcal{E}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})\right)\\
=\,&c_t\nu^2\cdot\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t}\right\|_{\mathrm{F}}^2\\
\leq\,&c_t\nu^2\cdot(M_t-k_t)+c_t\nu^2\cdot\frac{t-1}{\gamma_t^2}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}.
\end{aligned}$$

The last inequality follows from Proposition 1. Combining the above finishes the proof. □

While in Theorem 3 bounds the average training MSE loss $\frac{1}{M_t}\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2$, an alternative is to give a bound on $\frac{1}{M_t}\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2$. The latter term evaluates the difference between the prediction of $\widetilde{\boldsymbol{W}}_t$ and the ground-truth $\boldsymbol{W}_t^*$ on training data $\boldsymbol{H}_{1:t}$. The difference between the two terms is that $\boldsymbol{Y}_{1:t}=\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}+\mathcal{E}_{1:t}$ is contaminated by noise. We bound $\frac{1}{M_t}\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2$ in the next result.

Theorem 4. *On top of the settings of Theorem 1, furthermore assume $\mathcal{E}_{1:t}$ consists of i.i.d. $\mathcal{N}(0,\nu^2)$ entries. Then the output $\widetilde{\boldsymbol{W}}_t=\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$ of our method (4) satisfies*

$$\begin{aligned}
\frac{1}{M_t}\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\leq 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\left(\frac{a_t}{M_t}+\frac{a_{t-1}(t-1)}{\gamma_tM_t}+\frac{a_{t-1}(t-1)^2}{\gamma_t^2M_t}\right)\\
+c_t\nu^2\cdot\left(\frac{k_t}{M_t}+\left(\frac{t-1}{\gamma_t^2M_t}+\frac{2}{\gamma_tM_t}\right)\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}\right).
\end{aligned}$$

Proof of Theorem 4. We have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\\
=\,&\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\\
=\,&\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})+\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\\
=\,&\left\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top-\boldsymbol{I}_{M_t})\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2+\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2.
\end{aligned}$$

The first term is identical to that of Theorem 3, and it remains to bound the second term:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{E}_{1:t}}\left\|\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\\
=\,&\mathbb{E}_{\mathcal{E}}\operatorname{tr}\left(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\mathcal{E}_{1:t}{}^\top\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right)\\
=\,&c_t\nu^2\cdot\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\right\|_{\mathrm{F}}^2\\
\leq\,&c_t\nu^2\cdot\left(\frac{t-1}{\gamma_t^2}+\frac{2}{\gamma_t}\right)\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}+c_t\nu^2\cdot k_t.
\end{aligned}$$

The last inequality follows from Lemma 5. □

Theorem 5. *On top of the settings of Theorem 2, furthermore assume both $\mathcal{E}_{1:t}$ and $\boldsymbol{\epsilon}$ consists of i.i.d. $\mathcal{N}(0,\nu^2)$ entries. Then the output $\widetilde{\boldsymbol{W}}_t=\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$ of our method (4) satisfies*

$$\mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h},\boldsymbol{\epsilon}}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h}-\boldsymbol{y}\right\|^2\leq 2\cdot\|\boldsymbol{W}_t^*\|_{\mathrm{F}}^2\cdot\mathbb{B}_t+c_t\nu^2\cdot\mathbb{V}_t+c_t\nu^2. \tag{15}$$

where $\mathbb{B}_t$ and $\mathbb{V}_t$ are defined in (10) and also shown below:

$$\begin{aligned}
\mathbb{B}_t&=\left\|\boldsymbol{\Lambda}-\frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\|\left(1+\frac{(t-1)^2}{\gamma_t^2}\right)+\left(\frac{a_t}{M_t}+\frac{a_{t-1}(t-1)}{\gamma_tM_t}+\frac{a_{t-1}(t-1)^2}{\gamma_t^2M_t}\right)\\
\mathbb{V}_t&=\left\|\boldsymbol{\Lambda}-\frac{1}{M_t}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right\|\cdot\frac{\left(\frac{1}{\gamma_t}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}+k_t\right)}{\mu_{k_t}(\widetilde{\boldsymbol{B}}_t\widetilde{\boldsymbol{B}}_t^\top)}\\
&\quad+\frac{k_t}{M_t}+\left(\frac{t-1}{\gamma_t^2M_t}+\frac{2}{\gamma_tM_t}\right)\cdot\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}.
\end{aligned}$$

Proof of Theorem 5. Recall the definition of $\boldsymbol{D}_t$ in (13). Note that $\boldsymbol{D}_t$ is a symmetric and positive semi-definite matrix.

Note that for any $\boldsymbol{W} \in \mathbb{R}^{c_t \times E}$ we have

$$\begin{aligned}\mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h},\boldsymbol{\epsilon}}\left[\|\boldsymbol{W}\boldsymbol{h}-\boldsymbol{y}\|^2\right] &= \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h},\boldsymbol{\epsilon}}\left[\|\boldsymbol{W}\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}-\boldsymbol{\epsilon}\|^2\right] \\ &= \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|\boldsymbol{W}\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] + c_t\nu^2.\end{aligned}$$

Denote by $\boldsymbol{I}_E$ the $E \times E$ identity matrix. With $\widetilde{\boldsymbol{W}}_t = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$ and $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^*\boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}$ we obtain

$$\begin{aligned}&\mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|\widetilde{\boldsymbol{W}}_t\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\ =&\ \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\ =&\ \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|(\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}+\mathcal{E}_{1:t})\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\ =&\ \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|(\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}+\mathcal{E}_{1:t})\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] \\ =&\ \mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}-\boldsymbol{W}_t^*\boldsymbol{h}\|^2\right] + \mathbb{E}_{\mathcal{E}_{1:t},\boldsymbol{h}}\left[\|\mathcal{E}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2\right] \\ =&\ \mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}_t^*(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top-\boldsymbol{I}_E)\boldsymbol{h}\|^2\right] + c_t\nu^2\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{h}\|^2\right]\end{aligned}$$

The rest of the proof is identical to that of Theorem 2. □

G Additional Theoretical Results

Given the weight $\widetilde{\boldsymbol{W}}_t := \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top$ computed by our continual implementation in Section 4, here we aim to derive upper bounds on the training MSE losses $\frac{1}{M_t}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2$ without the linear model assumption $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^*\boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}$ as used in the main paper.

First observe that

$$\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t}-\boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 = \left\|\boldsymbol{Y}_{1:t}(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})\right\|_{\mathrm{F}}^2,$$

where we recall $\boldsymbol{I}_{M_t}$ is the $M_t \times M_t$ identity matrix. This motivates us to give a bound on $\|(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t})\|_{\mathrm{F}}^2$:

Proposition 1. *It holds for every $t \geq 1$ that ($M_0 := 0, k_0 := 0$)*

$$\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t}\right\|_{\mathrm{F}}^2 \leq M_t - k_t + \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}.$$

Remark 9. The term $M_t - k_t$ is inevitable as we truncate $M_t - k_t$ eigenvalues. Indeed, $M_t - k_t$ is precisely equal to $\|\boldsymbol{H}_{1:t}^\top\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}-\boldsymbol{I}_{M_t}\|_{\mathrm{F}}^2$, and it is the minimum of a rank-k_t approximation problem:

$$M_t - k_t = \min_{\boldsymbol{L}\in\mathbb{R}^{M_t\times k_t}} \|\boldsymbol{L}\boldsymbol{L}^\top - \boldsymbol{I}_{M_t}\|_{\mathrm{F}}^2.$$

The term $\frac{t-1}{\gamma_t^2}\min\left\{M_{t-1}-k_{t-1},(t-1)k_t\right\}$ arises as we solve ICL-TSVD continually rather than offline. With $\gamma_t = 1$, this term is upper bounded by $(t-1)(M_{t-1}-k_{t-1})$. With $\gamma_t = 10^{10}$ (as discussed in the main paper), this term is negligible for even hundreds of tasks.

Proof of Proposition 1. From Lemma 1 it follows that

$$\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top = \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top + \boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top, \tag{16}$$

where we recall $\boldsymbol{D}_t$ is defined as $\boldsymbol{D}_t = \sum_{i=1}^t \left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}\left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top\right)\right)$ in (13). Then we have

$$
\begin{aligned}
&\left\|\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - \boldsymbol{I}_{M_t}\right\|_{\mathrm{F}}^2 \\
&= \operatorname{tr}\left(\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} - 2\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t} + \boldsymbol{I}_{M_t}\right) \\
&= \operatorname{tr}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - 2\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) + M_t \\
&\overset{(16)}{=} \operatorname{tr}\left(\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top + \boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top + \boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right)\right) + M_t \\
&\quad - 2\left(\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top + \boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) \\
&= \operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) + M_t \\
&= \operatorname{tr}\left(\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\boldsymbol{D}_t\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right) + M_t - k_t \\
&\overset{\text{(i)}}{\leq} \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\} + M_t - k_t
\end{aligned}
$$

where (i) is due to Lemma 5. □

A simple corollary of Proposition 1 now follows:

Corollary 1. *The output $\widetilde{\boldsymbol{W}}_t$ of Algorithm 4 satisfies*

$$
\frac{1}{M_t}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \leq \frac{\|\boldsymbol{Y}_{1:t}^\top\boldsymbol{Y}_{1:t}\|}{M_t}\left(M_t - k_t + \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\}\right).
$$

Remark 10. In classification, the columns of $\boldsymbol{Y}_{1:t}$ are one-hot vectors. Hence, up to permutation, $\boldsymbol{Y}_{1:t}^\top\boldsymbol{Y}_{1:t} \in \mathbb{R}^{M_t\times M_t}$ is a block diagonal matrix with c_t block, where the i-th diagonal block is a $n_i \times n_i$ matrix of all ones $\mathbf{1}_{n_i}$; here n_i is the number of labels in class i. In other words, there exists a permutation matrix $\boldsymbol{\Pi}$ such that

$$
\boldsymbol{Y}_{1:t}^\top\boldsymbol{Y}_{1:t} = \boldsymbol{\Pi}\operatorname{diag}(\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \dots, \mathbf{1}_{n_{c_t}})\boldsymbol{\Pi}^\top.
$$

Since the maximum eigenvalue of $\mathbf{1}_{m_i}$ is m_i, we know

$$
\|\boldsymbol{Y}_{1:t}^\top\boldsymbol{Y}_{1:t}\| = \max_{i=1,\dots,c_t}\{n_i\}.
$$

Substitute this into Corollary 1 and we obtain

$$
\frac{1}{M_t}\left\|\widetilde{\boldsymbol{W}}_t\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\right\|_{\mathrm{F}}^2 \leq \frac{\max_{i=1,\dots,c_t}\{n_i\}}{M_t}\left(M_t - k_t + \frac{t-1}{\gamma_t^2}\min\left\{M_{t-1} - k_{t-1}, (t-1)k_t\right\}\right).
$$

It is also of interest to bound the distances between the SVD factors computed online and offline, namely the distances between $\widetilde{\boldsymbol{\Sigma}}_{1:t}, \overline{\boldsymbol{\Sigma}}_{1:t}$ and between $\widetilde{\boldsymbol{U}}_{1:t}, \overline{\boldsymbol{U}}_{1:t}$. We do so in the next result.

Theorem 6. *Let a_t be defined as in (6). For $t \geq 1$ define*

$$
\operatorname{gap}_t := \mu_{k_t}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right) - \mu_{k_t+1}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right). \tag{17}
$$

Then it always holds that

$$
\left\|\overline{\boldsymbol{\Sigma}}_{1:t}^2 - \widetilde{\boldsymbol{\Sigma}}_{1:t}^2\right\|_\infty \leq a_{t-1}. \tag{18}
$$

Moreover, if $a_{t-1} < \left(1 - 1/\sqrt{2}\right)\operatorname{gap}_t$, then for any $t \geq 1$ we have

$$
\min_{\boldsymbol{O}\in\mathcal{O}(k)}\left\|\overline{\boldsymbol{U}}_{1:t} - \widetilde{\boldsymbol{U}}_{1:t}\boldsymbol{O}\right\|_{\mathrm{F}} \leq \left\|\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\| \leq \frac{\sqrt{2}a_{t-1}}{\operatorname{gap}_t}, \tag{19}
$$

where $\mathcal{O}(k)$ be the set of $k \times k$ orthogonal matrices, defined as

$$
\mathcal{O}(k) := \{\boldsymbol{O} \in \mathbb{R}^{k\times k} : \boldsymbol{O}^\top\boldsymbol{O} = \boldsymbol{O}\boldsymbol{O}^\top = \boldsymbol{I}_k\}.
$$

Proof of Theorem 6. It is clear that $\overline{\boldsymbol{U}}_1 = \widetilde{\boldsymbol{U}}_1$ and $\overline{\boldsymbol{\Sigma}}_1 = \widetilde{\boldsymbol{\Sigma}}_1$. We now consider the case $t \geq 2$. Note that $\widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top$ is the eigen decomposition of $\tau_{k_t}(\widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{\Sigma}}^2_{1:t-1}\widetilde{\boldsymbol{U}}^\top_{1:t-1} + \boldsymbol{H}_t\boldsymbol{H}_t^\top)$, and $\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}\overline{\boldsymbol{U}}^\top_{1:t}$ is the eigen decomposition of $\tau_{k_t}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top)$. We can compute

$$\begin{aligned}\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top - \left(\widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{\Sigma}}^2_{1:t-1}\widetilde{\boldsymbol{U}}^\top_{1:t-1} + \boldsymbol{H}_t\boldsymbol{H}_t^\top\right) &= \boldsymbol{H}_{1:t-1}\boldsymbol{H}_{1:t-1}^\top - \widetilde{\boldsymbol{U}}_{1:t-1}\widetilde{\boldsymbol{\Sigma}}^2_{1:t-1}\widetilde{\boldsymbol{U}}^\top_{1:t-1} \\ &\overset{\text{(i)}}{=} \sum_{i=1}^{t-1}\left(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top - \tau_{k_i}(\widetilde{\boldsymbol{B}}_i\widetilde{\boldsymbol{B}}_i^\top)\right) =: \boldsymbol{D}_t,\end{aligned}$$

where (i) follows from Lemma 1. We can therefore apply Weyl's inequality to obtain

$$\left\|\overline{\boldsymbol{\Sigma}}^2_{1:t} - \widetilde{\boldsymbol{\Sigma}}^2_{1:t}\right\|_\infty \leq \|\boldsymbol{D}_t\| = a_{t-1}.$$

This proves (18). On the other hand, (19) follows from the Davis-Kahan theorem (Davis & Kahan, 1970), or more precisely, from Corollary 2.8 of Chen et al. (2021). □

H Review of Related Works

In Appendix H.1 we review related work on CL. Recent surveys on CL include Parisi et al. (2019); van de Ven et al. (2022); Shaheen et al. (2022); Zhou et al. (2024c); Wang et al. (2024); Shi et al. (2024). See also the GitHub repo of Liu (2024) for an extensive list of CL papers.

In Appendix H.2 we review related work on random feature models.

H.1 More Related Work on Continual Learning

Many CL methods have been proposed without explicitizing the use of pre-trained models (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Zeng et al., 2019; Chaudhry et al., 2019; Yan et al., 2021; Saha et al., 2021; Douillard et al., 2022). An easy way to boost their performance is to adapt them for the context of pre-trained models. There are two natural approaches to do so: (1) use the pre-trained model as initialization and run these CL algorithms to fine-tune the pre-trained model; or (2) train a shallow network with the output features of the pre-trained model and either of these CL algorithms. We do not explore these directions here. In what follows, we review existing CL methods designed for leveraging pre-trained models, and we review theoretical developments for CL as well.

Prior Work on CL with Pre-trained Models. The availability of pre-trained models has motivated new insights into designing CL methods. CL methods such designed can be roughly divided into two categories. In one category, the pre-trained model is completely frozen, and their output features are used as inputs for a tailored CL method. A straightforward method in this category, known as *SimpleCIL* (Zhou et al., 2023) or *Nearest Mean Classifier* (NMC) (Mensink et al., 2013; Rebuffi et al., 2017; Panos et al., 2023; Janson et al., 2022), is to classify a test image based on the (cosine) distances of its feature to class means of the training features. While this method is stable, hyperparameter-free, and can handle long task sequences, to the best of our knowledge, it does not have theoretical guarantees. Of course, RanPAC and our ICL-TSVD method also belong to this category.[1] Other methods in the category include Ahrens et al. (2024); Prabhu et al. (2024). Both methods make certain modifications on top of RanPAC:

- The method of Ahrens et al. (2024) replaces the random ReLU features with the concatenation of the output features of intermediate layers. We identify that this generalizes the idea of Pao & Takefuji (1992)[2].
- The method of Prabhu et al. (2024) replaces the random ReLU features with random Fourier features (which were used by Rahimi & Recht (2007) for learning kernel machines), and

[1] Note that RanPAC might use first-session adaptation (Remark 3), which modifies the output features of the pre-trained model, so one might not consider RanPAC as completely freezing the pre-trained model. However, such strategy of first-session adaptation is applied only before the first task, and is not used during continual learning of tasks. In other words, the model after first-session adaptation is completely frozen, and we might just view it as our pre-trained model.

the ridge regression solver of RanPAC with *linear discriminant analysis* (LDA) (Hayes & Kanan, 2020). Note that LDA optimizes an objective that is in general different from the MSE training loss, for which Prabhu et al. (2024) have not provided theoretical guarantees.

Similarly to RanPAC, the methods of Ahrens et al. (2024); Prabhu et al. (2024) need $O(E^3)$ time per task to invert the (regularized) $E \times E$ covariance matrix. Prabhu et al. (2024) uses $E = 25000$ in their experiments, which might constitute the current computational limit of performing the inversion (cf. Figs. 5 and 6). Also, both methods lack theoretical guarantees. In contrast, the running time of our ICL-TSVD method depends only linearly on E and can handle $E \geq 10^5$ with stable performance and estimation and generalization guarantees.

In the other category of methods, the weights of the pre-trained models remain fixed, but the output features of the pre-trained models are changed. The catch is that these methods either change the input or change the network architecture. Such change could be applied layer-wise, therefore, in order to describe the idea, it is the simplest to assume the pre-trained model f is a single-layer network.

- If we keep both the input and architecture fixed, then the network would take an input $\boldsymbol{X}$ and output $f(\boldsymbol{X})$.
- A popular way to change the input is to stack some trainable parameters $\boldsymbol{Z}$ with input $\boldsymbol{X}$, where $\boldsymbol{Z}$ and $\boldsymbol{X}$ have the same number of columns. The network outputs $f([\boldsymbol{X};\boldsymbol{Z}])$. Here, it is implicitly assumed that f can take input matrices with different number of rows (i.e., different number of tokens). For instance, f can be a single-layer vision transformer. In this case $\boldsymbol{Z}$ is often called (visual) *prompts* (Jia et al., 2022), and CL methods using this strategy are often called *prompt-based methods*; see, e.g., (Wang et al., 2022b;c;a; Smith et al., 2023; Wang et al., 2023; Jung et al., 2023; Tang et al., 2023; Gao et al., 2024b; Roy et al., 2024; Kim et al., 2024).
- A popular way to change the architecture is to replace the input-output map $\boldsymbol{X} \mapsto f(\boldsymbol{X})$ with $\boldsymbol{X} \mapsto f(\boldsymbol{X}) + g(\boldsymbol{X})$, where g is some simple shallow network parametrized by the extra trainable parameters. For instance, g could be a simple two-layer linear network of the form $g(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{B}\boldsymbol{X}$ or $g(\boldsymbol{X}) = \boldsymbol{A}\,\mathrm{relu}(\boldsymbol{B}\boldsymbol{X})$, where $\boldsymbol{A}, \boldsymbol{B}$ are trainable. In these case, $\boldsymbol{A}, \boldsymbol{B}$ are called *adapters* (Houlsby et al., 2019; Hu et al., 2022; Chen et al., 2022), and CL methods using this strategy are often called *adapter-based methods* (Zhou et al., 2023; 2024b; Liang & Li, 2024; Tan et al., 2024; Gao et al., 2024a).

Clearly, prompt-based and adapter-based methods can both be viewed as *expansion-based methods* that enlarge the capacity of a network in order to learn new tasks (Rusu et al., 2016; Yoon et al., 2018; Li et al., 2019; Ramesh & Chaudhari, 2022).

Despite their popularity, both prompt-based and adapter-based methods need to solve highly non-convex training problems, for which deriving informative theoretical guarantees is a significant challenge. Their lack of theoretical guarantees makes them prone to unexpected failures. For example, prompt-based methods such as *L2P* (Wang et al., 2022b), *DualPrompt* (Wang et al., 2022c), *CodaPrompt* (Smith et al., 2023), have their performance highly sensitive to the choice of hyperparameters and therefore to the pre-trained model in use (Wang et al., 2023), dataset, and problem setting (cf. Table 1); indeed, a small perturbation in learning rates might change the accuracy drastically (Zhang et al., 2023). While they are often equipped with dataset-specific hyperparameters released by authors (cf. Appendix J), their instability still emerges when applied to a long sequence of tasks. This is because new prompts or adapters are often needed to maintain high accuracy on new tasks (Zhou et al., 2024b), but doing so eventually becomes infeasible. Indeed, to train on the 100 tasks of the CIFAR100 dataset in the CIL setting with one class given at a time (B-0, Inc-1), running the adapter-based method of Zhou et al. (2024b), called *EASE*, with default hyperparameters, would create more than 117M parameters for its growing number of adapters, while the pre-trained ViTs in use have less than 87M parameters.

In Table 5 we summarize the conceptual differences of our approach from prior works.

Prior Work on CL Theory. Theoretical developments on CL have been chasing the current CL practice, with a majority of the theory CL papers limiting themselves to the linear, two-layer, or kernel setting (Doan et al., 2021; Heckel, 2022; Evron et al., 2022; Peng & Risteski, 2022; Lin et al., 2023; Swartworth et al., 2023; Goldfarb et al., 2024; Zhao et al., 2024; Ding et al., 2024). While these works cover various theoretical aspects (e.g., generalization bounds, sample complexity, and convergence

Table 5: Conceptual comparison to prior work.

	Estimation Guarantees?	Generalization Guarantees?	Stable?	Can Handle Long Task Sequences?
L2P, Dual Prompt, CodaPrompt	None	None	No	No
EASE	None	None	No	No
SimpleCIL	None	None	Yes	Yes
RanPAC	None	None	No	No
ICL-TSVD (Ours)	Theorem 1	Theorem 2	Yes	Yes

rates), there has arguably been a huge gap between their simplified settings and deep networks that state-of-the-art CL methods use. On the other hand, we have seen that deep pre-trained models in cascade with shallow trainable networks can provide competitive performance, thus it now makes sense to revise and extend these theoretical contributions within this cascaded architecture, thereby providing meaningful guarantees for learning the shallow networks. We believe this viewpoint would greatly reduce the gap between the theory and practice in the current CL literature.

H.2 Random Vector Functional Link Network and Random Feature Models

Here we make elaborations on Remarks 1, 4 and 6 and review related works mentioned therein. Recall our model is a two-layer network of the form

$$\boldsymbol{X} \mapsto \boldsymbol{W} \cdot \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}) \tag{20}$$

where $\boldsymbol{P}$ is randomly generated and fixed, and $\boldsymbol{W}$ consists of trainable parameters.

Random Vector Functional Link Network. In independent efforts, Schmidt et al. (1992) and Pao & Takefuji (1992) considered models of form (1). Schmidt et al. (1992) used the sigmoid activation function $\xi \mapsto \frac{1}{1+\exp(-\xi)}$, while Pao & Takefuji (1992) specified an arbitrary activation function as inspired by Hornik et al. (1989) and stack the features $\boldsymbol{X}$ and $\boldsymbol{H}$ together (see, e.g., Section 2.1 of Malik et al. (2023)).[2] The model Pao & Takefuji (1992) proposed has been known as *random vector functional link* (RVFL), and the model of Schmidt et al. (1992) is referred to, according to a recent review (Malik et al., 2023), as *Schmidt neural network* (SNN).

Models of form (20) are best combined with MSE losses, as training $\boldsymbol{W}$ with $\boldsymbol{P}$ fixed amounts to solving a least-squares problem, which admits a closed-form solution as shown by Schmidt et al. (1992) and even earlier by Webb & Lowe (1990).

The model Schmidt et al. (1992) and Pao & Takefuji (1992) advocated was proposed, again, by Huang et al. (2004; 2006) under the name *extreme learning machine* (ELM). While Huang et al. (2004) claimed ELMs to be a *new learning scheme* in the paper title, it was criticized by (Wang & Wan, 2008; Authors) that ELMs are ideas stolen from the last century (Schmidt et al., 1992; Pao & Takefuji, 1992), which Huang et al. (2004) were aware of yet did not cite. Despite the criticism, and perhaps because of its “fancy” name, ELMs had once been popular and attracted many follow-up variants. We shall not review these variants here.

The model we considered, therefore, follows in spirit the framework put forth by Schmidt et al. (1992); Pao & Takefuji (1992). Crucially, our approach is a modern instantiation of their framework (cf. Table 6), where we consider larger-scale problems with ill-conditioned data, online solvers with GPU implementations. But does it make sense to use the random ReLU features $\boldsymbol{H}_t := \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$ rather than the pre-trained features $\boldsymbol{X}_t$ for regression? Would the transformation $\boldsymbol{P} \in \mathbb{R}^{E\times d}$ even harm the pre-trained knowledge? McDonnell et al. (2023) empirically verified that having the first layer $\boldsymbol{P}$ is beneficial to performance as long as $E \geq d$, and the accuracy tends to be higher for larger E (see Table A5 of Appendix F.6 of McDonnell et al. (2023)). While RanPAC is limited to $E \approx 10^4$, our approach is inherently more scalable, allowing us to take $E = 10^5$. The pursuit in higher embedding dimension E brings us into the *over-parameterized* territory, where the corresponding MSE objective has infinitely many solutions, and this is different from the classic works on RVFLs or SNNs that largely focus on the case where there are just a few hundred neurons (e.g., $E \approx 100$).

[2] Hence, the method of Ahrens et al. (2024) can be viewed as a modern variant of Pao & Takefuji (1992) as Ahrens et al. (2024) stacks the output features of multiple intermediate layers for regression.

Table 6: Comparison between our approach and classic methods for extreme learning machines.

	Problem Scale	Solver	Data	Compute Platform
Schmidt et al. (1992); Pao & Takefuji (1992)	small	offline	well-conditioned	CPU
ICL-TSVD (Ours)	large	online	ill-conditioned	GPU

Random Feature Models. Model (20) is also studied under the name *random feature model* (RFM) with the origin of RFMs often attributed to Rahimi & Recht (2007). The RFM is considered to be a simple proxy model for understanding deep networks, and hence it has recently been popular (Belkin et al., 2018; 2019; Bartlett et al., 2020; Hastie et al., 2022; Mei & Montanari, 2022; Tsigler & Bartlett, 2023). Some of these works make statistical assumptions on $\boldsymbol{X}_t$ and address technical challenges in analyzing the nonlinear map $\boldsymbol{W} \cdot \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$.

Alternatively, one could conduct analysis conditioned on $\boldsymbol{H}_t := \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$, which would be more manageable as it reduces to linear models. For example, the work of Xu & Hsu (2019); Huang et al. (2022); Bach (2024); Green & Romanov (2024) truncates the SVD factors of the features before applying least-squares. This is similar to ours, with an important difference that they apply TSVD only once, while we apply it continually. Also, their results make statistical (e.g., Gaussian) assumptions on $\boldsymbol{H}_{1:t}$, which could violate our context that $\boldsymbol{H}_{1:t}$ is generated via $\boldsymbol{H}_t := \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$. In contrast, our results in the main paper, namely Theorems 1 and 2, make no assumptions on $\boldsymbol{H}_{1:t}$ and are therefore applicable to random feature models and to the pre-trained models bridged with a random feature model.

I Dataset Details

For convenience and completeness, we collect some details about the datasets in Tables 7 and 8.

Table 7: Datasets used for class-incremental learning experiments. †: ObjectNet, OmniBenchmark, and VTAB contain a large number of classes, and we use a subset of these datasets delivered by Zhou et al. (2023); see their Table 4 and also Table A2 of McDonnell et al. (2023).

Dataset Name	Origin	Training Set Size	Test Set Size	# of Classes	Link
CIFAR100	(Krizhevsky et al., 2009)	50,000	10,000	100	Here
ImageNet-R	(Hendrycks et al., 2021a)	24,000	6,000	200	Here
ImageNet-A	(Hendrycks et al., 2021b)	5,981	5,985	200	Here
CUB-200	(Wah et al., 2011)	9,430	2,358	200	Here
ObjectNet†	(Barbu et al., 2019)	26,509	6,628	200	Here
OmniBenchmark†	(Zhang et al., 2022)	89,697	5,985	300	Here
VTAB†	(Zhai et al., 2019)	1,796	8,619	50	Here
StanfordCars	(Krause et al., 2013)	8,144	8,041	196	Here

Table 8: Datasets used for domain-incremental learning. See Table A3 of McDonnell et al. (2023) for even more details.

Dataset Name	Origin	Training Set Size	Test Set Size	# of Classes	Link
CORe50	(Lomonaco & Maltoni, 2017)	119,894	44,972	50	Here
CDDB-Hard	(Li et al., 2023)	16,068	5,353	2	Here
DomainNet	(Peng et al., 2019)	409,832	176,743	345	Here

J Experimental Setup Details

The details of how we run each of the methods are specified as follows. For L2P, DualPrompt, CodaPrompt, we use the hyperparameters available in the PILOT repo (Sun et al., 2023) for the CIFAR100 and ImageNet-R datasets. Since no official hyperparameters are released for other datasets, we simply use their respective hyperparameters of CIFAR100 for other datasets. One might notice that these methods have large accuracy drops on other datasets, suggesting that they are sensitive to hyperparameters and the good and dataset-specific hyperparameters, if they exist, are to be found for these methods to perform well. This is an inherent drawback as they involve minimizing a highly non-convex training objective. On the other hand, many other methods, including SimpleCIL, RanPAC, and ours, are almost parameter-free. Specifically, the only hyperparameter of our approach is the truncation threshold and its role is clearly explained in the main paper.

For joint linear classifiers, that is LC ($\boldsymbol{X}_{1:T}$) or LC ($\boldsymbol{H}_{1:T}$), we train for 20 epochs using the cross-entropy loss, batch size 48, weight decay 0.0005, and SGD with the cosine annealing schedule. We run LC ($\boldsymbol{X}_{1:T}$) and LC ($\boldsymbol{H}_{1:T}$) with different initial learning rates $\{0.001, 0.005, 0.01, 0.02, 0.03\}$, and take report the maximum accuracy (Table 9). Note that LC ($\boldsymbol{X}_{1:T}$) and LC ($\boldsymbol{H}_{1:T}$) are trained using the cross-entropy loss, not the MSE loss. The reason is that the features $\boldsymbol{H}_{1:t}$ are highly ill-conditioned (Fig. 1), which makes SGD converge very slowly with the MSE loss. Comparing this to Remark 2, we conclude that the MSE loss in our setting is useful when the objective is minimized via robust numerical computation techniques (e.g., our TSVD implementation in Appendix C) instead of SGD.

We also consider the idea of *first-session adaptation* (cf. Remark 3). This idea introduces a few hyperparameters such as the learning rate and schedule. We run experiments with two sets of hyperparameters, given respectively by RanPAC and EASE. We attach the symbol † to the method name when we use the hyperparameters of RanPAC (e.g., ADAM†, ICL-TSVD†, RanPAC†). We attach the symbol $*$ when we use the hyperparamters of EASE (e.g., ICL-TSVD*, EASE*).

For RanPAC, the official hyperparameters given by McDonnell et al. (2023) vary for different datasets. We try to unify the setup by keeping using the hyperparameters most frequently used by McDonnell

Table 9: Accuracy of training joint linear classifiers given all data with different initial learning rates $\{0.001, 0.005, 0.01, 0.02, 0.03\}$ and the corresponding maximum accuracy. LC ($\boldsymbol{X}_{1:T}$) trains a linear classifier using all pre-trained features $\boldsymbol{X}_{1:T}$ and LC ($\boldsymbol{H}_{1:T}$) uses all embedded features.

	LC ($\boldsymbol{X}_{1:T}$)						LC ($\boldsymbol{H}_{1:T}$)					
	0.001	0.005	0.01	0.02	0.03	Max	0.001	0.005	0.01	0.02	0.03	Max
ViTs pre-trained on ImageNet-1K (`vit_base_patch16_224`):												
CIFAR100	86.39	87.56	87.47	87.09	86.99	87.56	87.76	86.68	86.36	86.75	86.46	87.76
ImageNet-R	70.25	72.42	72.22	72.02	71.52	72.42	73.00	71.08	71.18	70.70	71.15	73.00
ImageNet-A	54.64	58.85	58.46	57.67	56.55	58.85	59.25	56.16	56.09	56.48	56.42	59.25
CUB-200	82.32	87.62	88.59	88.63	88.76	88.76	88.72	88.13	88.08	88.17	87.83	88.72
ObjectNet	57.53	59.70	59.22	58.68	58.43	59.70	59.96	56.14	55.63	55.48	55.34	59.96
Omnibenchmark	76.11	79.00	79.55	79.43	79.50	79.55	80.02	79.62	79.57	79.62	79.43	80.02
VTAB	86.40	90.89	91.32	91.23	90.89	91.32	91.17	89.86	89.99	90.16	89.99	91.17
StanfordCars	39.56	62.54	69.29	72.86	74.12	74.12	72.43	73.65	73.54	72.81	72.49	73.65
ViTs pre-trained on ImageNet-21K (`vit_base_patch16_224_in21k`):												
CIFAR100	86.15	86.78	86.33	85.86	85.17	86.78	85.80	85.16	85.05	85.31	85.4	85.80
ImageNet-R	67.22	68.63	67.12	65.90	65.28	68.63	68.65	68.00	68.17	68.23	67.78	68.65
ImageNet-A	46.68	51.42	50.03	49.24	48.58	51.42	51.15	50.16	49.44	48.98	49.64	51.15
CUB-200	85.58	88.89	89.06	88.72	88.21	89.06	89.31	88.17	88.63	88.46	88.51	89.31
ObjectNet	58.01	58.39	57.50	55.87	54.42	58.39	56.93	53.33	54.44	54.47	54.54	56.93
Omnibenchmark	78.95	79.67	79.73	79.45	79.33	79.73	79.62	79.26	79.11	78.91	79.05	79.62
VTAB	87.71	90.78	90.97	90.71	90.11	90.97	90.78	91.03	90.42	90.44	90.27	91.03
StanfordCars	44.80	64.22	68.06	68.92	68.71	68.92	69.38	67.64	67.65	67.79	67.39	69.38

et al. (2023). Specifically, we set the embedding dimension E to 10000 for RanPAC; note the exception that McDonnell et al. (2023) run the CDDB experiments with $E = 5000$, even though their Table A5 showed that larger E in general leads to higher accuracy on CIFAR100. The main hyperparameters of RanPAC used for first-session adaptation are as follows:

```
{"tuned_epoch":20,
 "init_lr":0.01,
 "batch_size":48,
 "weight_decay":0.0005}
```

We run ADAM†, ICL-TSVD†, RanPAC† where first-session adaptation uses these hyperparameters consistently for all datasets.

The EASE approach of Zhou et al. (2024b) performs fine-tuning, not just for the first session, but for every session, in an interesting way. The hyperparameters in their released code vary for different sessions and different datasets, and we refer the reader to the official GitHub repo of EASE for details. We also run ICL-TSVD* with the hyperparameters of EASE for first-session adaptation. Note that since EASE does not show experiments on StanfordCars, or does not release hyperparameters on this dataset, we run EASE with its CIFAR100 hyperparameters for StanfordCars; see also Table 10 of Appendix K.1 where we tune the initial learning rates of EASE on StanfordCars, showing that the accuracy is still low.

Finally, we note that in all tables, some approaches are marked in gray; they are not directly comparable to our approach as the methodology can be very different and it is in fact possible to combine one with another for even better performance. On the other hand, RanPAC is the most related to our method, hence we highlight the comparison with the purple background.

K Extra Experiments, Figures, and Tables

K.1 Performance on StanfordCars

It is observed in Table 1 that many methods exhibit significant performance drops on StanfordCars. Are these methods inherently unable to handle this dataset, or is it our taking inappropriate hyperparameters that lead to poor performance? Note that the authors of these works did not test their

Table 10: Accuracy of EASE with different initial learning rates $\{0.001, 0.005, 0.01, 0.02, 0.03\}$ on StanfordCars.

0.001			0.005			0.01			0.02			0.03		
Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20
33.42	32.27	19.38	33.48	32.33	19.46	33.39	32.65	20.06	32.58	32.11	24.96	32.41	31.77	29.76

Table 11: Final accuracy of different methods using ViTs pre-trained on ImageNet-21K (`vit_base_patch16_224_in21k`). Compare this with Table 1 of the main paper.

(Part 1)	CIFAR100 (B-0)			ImageNet-R (B-0)			ImageNet-A (B-0)			CUB-200 (B-0)			Avg.
LC ($\boldsymbol{X}_{1:T}$)	86.78			68.63			51.42			89.06			73.97
LC ($\boldsymbol{H}_{1:T}$)	85.80			68.65			51.15			89.31			73.73
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
RanPAC	87.58	87.65	87.75	68.18	70.03	70.13	**37.59**	52.01	52.73	89.48	89.65	89.44	73.52
ICL-TSVD	88.62	88.63	88.67	70.85	70.93	70.72	**54.71**	54.71	55.10	90.33	90.33	90.46	76.17
EASE*	85.85	87.67	89.47	70.27	74.53	75.88	43.05	47.53	54.51	86.77	86.81	85.50	73.99
RanPAC*	90.26	91.39	91.97	75.47	76.10	77.33	**47.60**	58.00	62.74	83.21	89.57	89.69	77.78
ICL-TSVD*	90.55	91.88	92.39	76.48	76.82	77.25	**56.95**	58.85	62.74	90.63	90.71	90.67	79.66
(Part 2)	ObjectNet (B-0)			OmniBenchmark (B-0)			VTAB (B-10)			StanfordCars (B-16)			Avg.
LC ($\boldsymbol{X}_{1:T}$)	58.39			79.73			90.97			68.92			74.50
LC ($\boldsymbol{H}_{1:T}$)	56.93			79.62			91.03			69.38			74.24
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
RanPAC	59.14	59.23	59.29	78.38	78.40	78.06	92.37	92.84	92.84	**43.60**	65.78	65.78	72.14
ICL-TSVD	61.35	61.36	61.33	80.12	80.03	80.13	92.90	92.76	92.92	**68.96**	68.76	68.95	75.80
EASE*	54.98	57.50	60.35	72.88	73.50	73.87	88.24	93.46	93.43	35.43	34.62	37.32	64.63
RanPAC*	64.56	66.55	66.05	78.55	79.15	79.23	92.77	92.84	93.72	**2.31**	69.11	69.11	71.16
ICL-TSVD*	66.76	66.67	66.93	80.55	80.82	81.55	93.85	93.79	93.76	**71.46**	71.58	71.71	78.29

methods on StanfordCars, nor they released the corresponding hyperparameters. To rule out the case of hyperparameter misspecification, we take the EASE* method of Zhou et al. (2024b) for example, and we run it with different initial learning rates $\{0.001, 0.005, 0.01, 0.02, 0.03\}$ for the first task (this hyperparameter is called `init_lr` in the JSON file of the code repo of Sun et al. (2023); all other hyperparameters are set to the corresponding hyperparamters Zhou et al. (2024b) gave for CIFAR100. The results are in Table 10. It shows that different initial learning rates for EASE do not improve the performance on StanfordCars too much, suggesting that StanfordCars is perhaps inherently difficult for these types of methods.

K.2 Experiments with ViT features Pre-trained on ImageNet-21K

Note that by default we use ViTs pre-trained on ImageNet-1K (`vit_base_patch16_224`). Here in Table 11 we show experiments with ViTs pre-trained on ImageNet-21K. Comparing this with Table 1, we obtain a similar conclusion that ICL-TSVD is more stable than and outperforms RanPAC.

K.3 Experiments on Domain-Incremental Learning (DIL)

Here we consider *domain-incremental learning* (DIL), where each task has images of all objects collected from different sources or domains, e.g., objects in the images of task 1 could be hand-written sketches of cars, and images of task 2 could be colored cars.

We follow the work of McDonnell et al. (2023) to run domain-incremental learning experiments on 3 datasets, CORe50 (Lomonaco & Maltoni, 2017), CDDB-Hard (Li et al., 2023), and DomainNet (Peng et al., 2019). The corresponding experimental results are shown in Table 12, from which we observe a similar phenomenon: ICL-TSVD is more stable and has higher accuracy.

Table 12: Final accuracies of RanPAC and ICL-TSVD with pre-trained ViTs for domain incremental learning.

	CORe50	CDDB-Hard	DomainNet	Avg.
RanPAC	94.98	75.14	64.20	78.11
ICL-TSVD	96.06	79.21	67.18	80.82

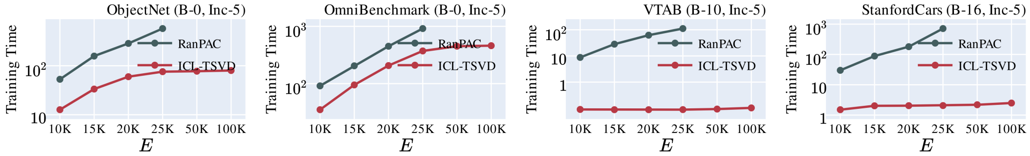


Figure 6: Training times of ICL-TSVD and RanPAC for different embedding dimensions E (in minutes). See also Fig. 5 in the main paper for similar results on the other four datasets.

K.4 Scaling Laws of the Embedding Dimension

Fig. 7 exhibits a scaling law where the accuracy of ICL-TSVD grows with the embedding dimension E. A similar phenomenon is also shown for RanPAC in Table A5 of Appendix F.6 of McDonnell et al. (2023). However, the experiments of McDonnell et al. (2023) are limited to $E = 15000$ as RanPAC is not scalable (cf. Figs. 5 and 6), and also limited to only the CIFAR100 dataset. Here, since our ICL-TSVD implementation is more stable and scalable, we can run it with E as large as 10^5, therefore visualizing the scaling phenomenon for eight different datasets in Fig. 7.

It is clearly tempting to scale the embedding dimension even more, but we have not found any significant performance gain with even larger E (e.g., $E = 150000$ or $E = 200000$), which is why we stopped at dimension $E = 10^5$. Note that enlarging E amounts to increasing the width of the corresponding layer. It was suggested that increasing the width is beneficial, theoretically (Peng et al., 2023) and empirically (Mirzadeh et al., 2022). On the other hand, Fig. 7 suggests the benefit of increasing the width empirically diminishes, e.g., the accuracies for $E = 50\text{K}$ and $E = 100\text{K}$ are comparable on ImageNet-A, CUB-200, VTAB, and StanfordCars. This is empirically corroborated by Guha & Lakshman (2024) and theoretically confirmed by Hu et al. (2024).

K.5 More Figures for Data Analysis

Recall $\mu_k(\cdot)$ denotes the k-th largest eigenvalue of a matrix and $M_t := m_1 + \cdots m_t$. Define the notion of *effective rank*, as by Tsigler & Bartlett (2023):

$$r_k\left(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}\right) := \frac{\sum_{j\geq k} \mu_j\left(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}\right)}{\mu_k\left(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}\right)}, \quad \forall k = 1, \dots, M_t \tag{21}$$

Note that $r_1(\cdot)$ is the standard definition of effective rank (sometimes called *stable rank*), while $r_k(\cdot)$ generalizes it by only considering the eigenvalues starting from the k largest.

Fig. 8a plots the effective rank $r_1(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t})$ as the number of tasks increases and Fig. 8b plots $r_k(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t})$ for different values of k. We observe similar curves on different datasets: $r_1(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t})$ is smaller than 3, and $r_k(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t})$ first increases and then decreases as a function of k.

Fig. 9 plots the top k_t eigenvalues of $\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top$ (recall that we only preserve top k_t singular values of $\widetilde{\boldsymbol{B}}_t$). It shows that the condition number is now of order 10^5 ($10^{11}/10^6$), which is much smaller than the condition number of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$.

Fig. 10 plots the normalized differences $\|\overline{\boldsymbol{\Sigma}}_{1:t} - \widetilde{\boldsymbol{\Sigma}}_{1:t}\|_\infty / \|\overline{\boldsymbol{\Sigma}}_{1:t}\|_\infty$ between the eigenvalues given by ICL-TSVD and its continual implementation (Algorithm 4). This empirically verifies that the differences between the two are insignificant compared to the largest eigenvalue $\|\overline{\boldsymbol{\Sigma}}_{1:t}\|_\infty$ (just of order 10^{-3}). This experiment assists understanding Theorem 6.

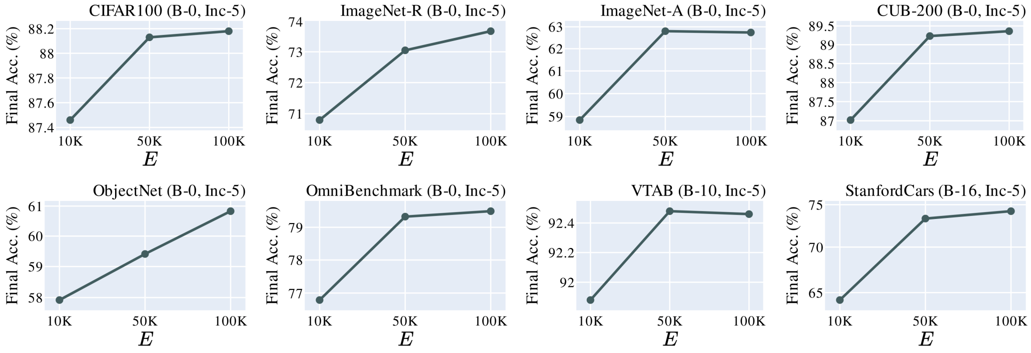


Figure 7: Final accuracy of ICL-TSVD for varying embedding dimensions E. See also Table 13 in comparison to RanPAC.

Table 13: Final accuracies of RanPAC and ICL-TSVD with pre-trained ViTs. RanPAC takes its default choice $E = 10\text{K}$, while for ICL-TSVD we set three different values for E: $E = 10\text{K}$, $E = 50\text{K}$, and $E = 100\text{K}$.

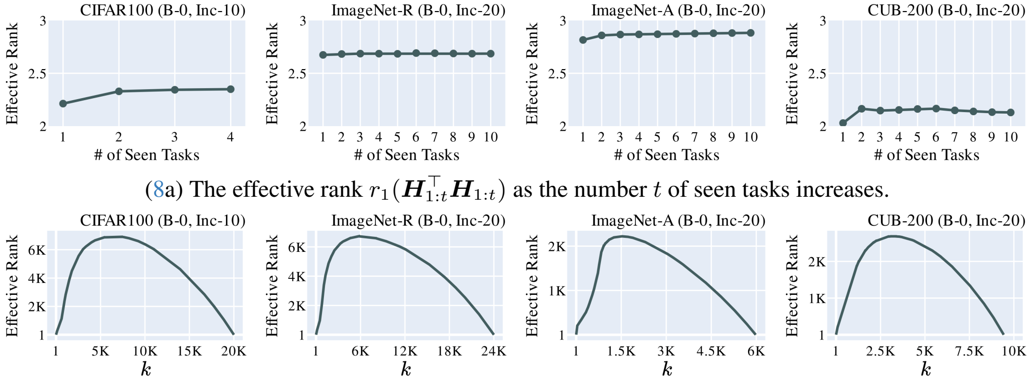
(Part 1)	CIFAR100 (B-0)			ImageNet-R (B-0)			ImageNet-A (B-0)			CUB-200 (B-0)			Avg.
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
RanPAC	86.71	87.02	87.10	71.90	71.97	72.50	56.48	62.34	61.75	88.08	87.15	88.13	76.76
Ours ($E = 10\text{K}$)	87.48	87.49	87.42	70.77	70.85	70.48	58.85	58.46	59.38	86.73	86.85	87.19	76.00
Ours ($E = 50\text{K}$)	88.13	88.05	88.04	73.05	73.07	73.05	62.80	62.80	62.48	89.23	89.23	89.19	78.26
Ours ($E = 100\text{K}$)	88.18	88.18	88.21	73.67	73.72	73.63	62.74	63.20	63.20	89.36	89.27	89.23	78.55
(Part 2)	ObjectNet (B-0)			OmniBenchmark (B-0)			VTAB (B-10)			StanfordCars (B-16)			Avg.
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	
RanPAC	58.77	57.66	57.69	77.63	77.63	77.46	91.15	91.58	91.58	58.03	71.40	71.40	73.50
Ours ($E = 10\text{K}$)	57.97	57.82	58.06	76.79	76.96	76.91	91.89	91.81	91.91	64.03	64.43	64.72	72.78
Ours ($E = 50\text{K}$)	59.41	59.4	59.54	79.33	79.35	79.43	92.48	92.47	92.54	73.32	73.66	73.47	76.20
Ours ($E = 100\text{K}$)	60.83	60.86	60.77	79.50	79.60	79.70	92.46	92.55	92.56	74.21	74.39	74.39	76.82

Fig. 11 plots the eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t} \in \mathbb{R}^{M_t \times M_t}$ ($M_t \leq 10^4$) with the embedding dimension E varying in $\{10000, 25000, 50000, 75000\}$. It shows that the "shape" of the spectrum is similar for different values of E and on different datasets (see also Fig. 1a for the case $E = 10^5$).

Fig. 12 depicts how the random embedding $\boldsymbol{P} \in \mathbb{R}^{E \times d}$ and ReLU layer affect the spectrum of the features. In Fig. 12a we plot the output features $\boldsymbol{X} \in \mathbb{R}^{d \times M}$ of the ImageNet-A dataset from pre-trained ViTs ($d = 768, M = 5981, E = 10^5$). We see $\boldsymbol{X}$ is relatively well-conditioned: Its maximum eignvalue is of order 10^5 and minimum eigenvalue of order 10. Fig. 12b shows that $\boldsymbol{PX} \in \mathbb{R}^{E \times M}$ has extremely small eigenvalues. This is because $\boldsymbol{PX}$ has rank at most d, and the smallest $M - d$ eigenvalues of $\boldsymbol{X}^\top \boldsymbol{P}^\top \boldsymbol{PX}$ should be zero, while we get these small and non-zero eigenvalues in Fig. 12b due to numerical errors in (incremental) SVD; these eigenvalues should be truncated (set to zero), in order to solve Min-Norm ICL accurately. Fig. 12c shows that $\operatorname{relu}(\boldsymbol{PX})$ also has these small and non-zero eigenvalues. While the rank of $\operatorname{relu}(\boldsymbol{PX})$ is unclear, its smallest yet non-zero eigenvalues are likely inherent from $\boldsymbol{PX}$, and we suggest truncating them as well. See also Table 14.

K.6 Ablation Study on Random ReLU Features

In Table 14 we study the effects of the random ReLU model. Recall that, given the output features $\boldsymbol{X}_t$ of the data of task t from a pre-trained model, we use the random ReLU features $\boldsymbol{H}_t := \operatorname{relu}(\boldsymbol{PX}_t)$ and labels $\boldsymbol{Y}_t$ to train a linear classifier via continually solving Min-Norm ICL or ICL-TSVD. We could instead use $\boldsymbol{X}_t$ or $\boldsymbol{PX}_t$ or $\operatorname{relu}(\boldsymbol{X}_t)$ as the features to train the linear classifier. To see the effects of these alternative choices, we make Table 14 from which we have the following observations:



(8a) The effective rank $r_1(\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t})$ as the number t of seen tasks increases.

(8b) The eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$ arranged in descending order (t fixed).

Figure 8: The effective ranks of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t}$.

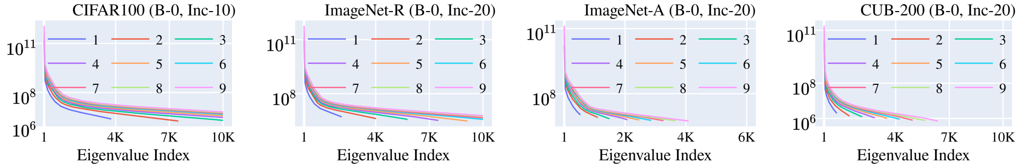


Figure 9: The eigenvalues of $\widetilde{\boldsymbol{\Sigma}}_{1:t}$ for $t = 1, \dots, 9$. These are by definition the top k_t eigenvalues of $\widetilde{\boldsymbol{B}}_t \widetilde{\boldsymbol{B}}_t^\top$. It shows that our continual implementation (Algorithm 4) prunes the extremely small eigenvalues of order 10^{-5} so that the condition number is now of order 10^5 ($10^{11}/10^6$); compare this figure with Fig. 1a. In this experiment we truncate 25% of the eigenvalues; that is, given M_t, we select k_t such that $k_t/M_t = 75\%$.

- The random ReLU features $\boldsymbol{H}_t$ gives the highest accuracy, while using the ReLU layer alone or random embedding alone does not make improvements over the original pre-trained features $\boldsymbol{X}_t$.
- Solving Min-Norm ICL via Incremental SVD exhibits numerical failures as soon as we use random embedding $\boldsymbol{P}$. This is because $\boldsymbol{P}\boldsymbol{X} \in \mathbb{R}^{E\times M}$ has rank at most d and we would get some $M - d$ small yet non-zero eigenvalues accounting for the numerical errors of (incremental) SVD solvers (see, e.g., Fig. 12). While they damage the accuracy, truncating these singular values and the corresponding singular vectors restores the performance.

K.7 Extra Empirical Study of RanPAC

In Appendix K.7.1, we analyze the training losses of RanPAC. In Appendix K.7.2 we show RanPAC is unstable with respect to small increments, while ICL-TSVD is more stable.

K.7.1 Training losses of RanPAC

Fig. 13 plots the training loss of RanPAC for different ridge regularization parameters $\lambda \in \{0, 1, 10, 100, 1000\}$. We observe that, In the case of $\lambda = 0$, the training loss of RanPAC is smaller than 1. Fig. 3 shows that the incremental SVD implementation of Min-Norm ICL could have its training loss larger than 10^{10}. This difference is because the incremental SVD implementation (without truncation) can be unstable and accumulates errors over time, while RanPAC is implemented by solving the normal equations $\boldsymbol{W}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \lambda \boldsymbol{I}_E) = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ in variable $\boldsymbol{W}$ (the covariances $\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top$ and $\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ are updated continually). The advantage is that maintaining $\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top$ and $\boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ is easy and does not entail numerical errors, so solving the normal equations $\boldsymbol{W}(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top + \lambda \boldsymbol{I}_E) = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top$ directly is expected to be stable, as long as the solver invoked is numerically stable (the built-in PyTorch solver is used). This appears to be the case, as RanPAC

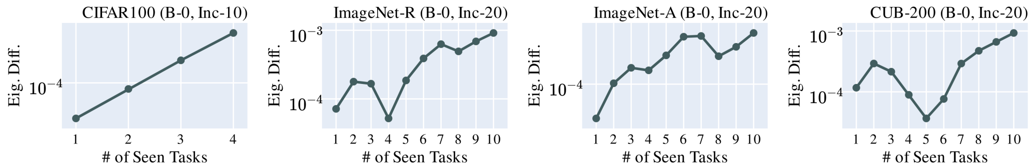


Figure 10: Normalized differences $\|\overline{\mathbf{\Sigma}}_{1:t} - \widetilde{\mathbf{\Sigma}}_{1:t}\|_\infty / \|\overline{\mathbf{\Sigma}}_{1:t}\|_\infty$ between the eigenvalues given by ICL-TSVD and its continual implementation (Algorithm 4). This empirically verifies that the differences between the two are insignificant compared to the largest eigenvalue $\|\overline{\mathbf{\Sigma}}_{1:t}\|_\infty$ (just of order 10^{-3}). See Fig. 9 and Fig. 1a. See also Theorem 6 where we formally bound the distances $\|\overline{\mathbf{\Sigma}}_{1:t} - \widetilde{\mathbf{\Sigma}}_{1:t}\|_\infty$ for every t.

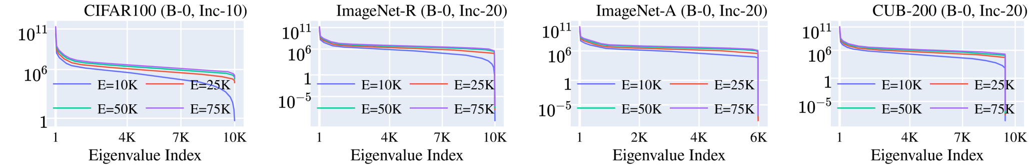


Figure 11: Eigenvalues of $\boldsymbol{H}_{1:t}^\top \boldsymbol{H}_{1:t} \in \mathbb{R}^{M_t \times M_t}$ ($M_t \leq 10^4$) with the embedding dimension E varying in $\{10000, 25000, 50000, 75000\}$. We find that the "shape" of the spectrum is similar for different E and on different datasets (see also Fig. 1a for the case $E = 10^5$).

maintains small training errors. On the other hand, the corresponding test accuracy can be nearly zero with small λ (e.g., when $\lambda = 0, 1$ as shown in Fig. 2).

K.7.2 RanPAC is Unstable for CIL with The Smallest Increments

In the experiments of the main paper, we see that RanPAC is unstable for small increments (e.g., Inc-5). Moreover, its instability is exacerbated in the extreme case Inc-1 (Table 2).

Here, we show similar experimental results in Figs. 14 to 17, suggesting that RanPAC is significantly more unstable for Inc-1, Inc-2, and Inc-4. In particular, in the extreme case of Inc-1, RanPAC presents failures on all datasets except CIFAR100 (as indicated by the verticle blue line in Figs. 14 to 17. On the contrary, these figures, including Fig. 18, show that our ICL-TSVD method is stable for different small increments (1, 2, 4, 5).

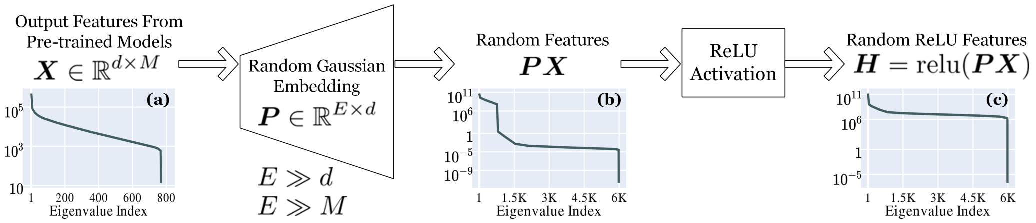


Figure 12: Feeding M data samples to a pre-trained model gives its d-dimensional output features $\boldsymbol{X} \in \mathbb{R}^{d \times M}$. Passing it through a random embedding layer $\boldsymbol{P}$ and a ReLU layer yields the random ReLU features $\boldsymbol{H} \in \mathbb{R}^{E \times M}$. Plotted in (a), (b), (c), respectively, are eigenvalues of $\boldsymbol{X}\boldsymbol{X}^\top$, $\boldsymbol{X}^\top \boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{X}$, and $\boldsymbol{H}^\top \boldsymbol{H}$ in descending order, where $\boldsymbol{X}$ consists of pre-trained ViT features of the ImageNet-A dataset ($d = 768, M = 5981, E = 10^5$). It is seen that $\boldsymbol{PX}$ and ,$\boldsymbol{H}$ are more ill-conditioned than $\boldsymbol{X}$. See also Table 14.

Table 14: Final accuracy of Min-Norm ICL and ICL-TSVD when using different features, $\boldsymbol{X}_t \in \mathbb{R}^{d\times m_t}$, $\mathrm{relu}(\boldsymbol{X}_t)$, $\boldsymbol{P}\boldsymbol{X}_t$, and $\boldsymbol{H}_t := \mathrm{relu}(\boldsymbol{P}\boldsymbol{X}_t)$. Here $\boldsymbol{P} \in \mathbb{R}^{E\times d}$ is a random Gaussian matrix with $\mathcal{N}(0,1)$ entries and $E = 10^5$, and $\boldsymbol{H}_t$ consists of random ReLU features we use by default. Note that both Min-Norm ICL and ICL-TSVD are solved by the incremental SVD method, with a difference that the latter truncates the SVDs. Incremental SVD without truncation is not scalable enough to handle all 50000 data samples of CIFAR100, so we mark "N.A." in the table for Min-Norm ICL.

	CIFAR100 (B-0, Inc-10)	ImageNet-R (B-0, Inc-20)	ImageNet-A (B-0, Inc-20)	CUB (B-0, Inc-20)	Avg.
		Final Accuracy of Min-Norm ICL			
$\boldsymbol{X}_t$	85.11	69.22	58.92	84.90	74.54
$\mathrm{relu}(\boldsymbol{X}_t)$	84.07	66.43	55.23	84.14	72.47
$\boldsymbol{P}\boldsymbol{X}_t$	N.A.	1.35	2.37	0.55	N.A.
$\boldsymbol{H}_t$	N.A.	0.42	0.92	0.72	N.A.
		Final Accuracy of ICL-TSVD			
$\boldsymbol{X}_t$	84.61	68.23	59.45	84.14	74.11
$\mathrm{relu}(\boldsymbol{X}_t)$	83.51	66.20	56.62	84.01	72.59
$\boldsymbol{P}\boldsymbol{X}_t$	78.96	48.50	42.73	63.15	58.33
$\boldsymbol{H}_t$ (default)	88.18	73.65	63.20	89.23	78.57

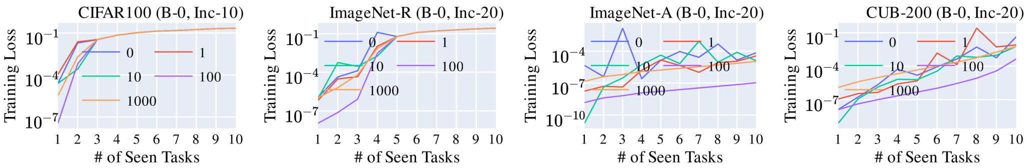
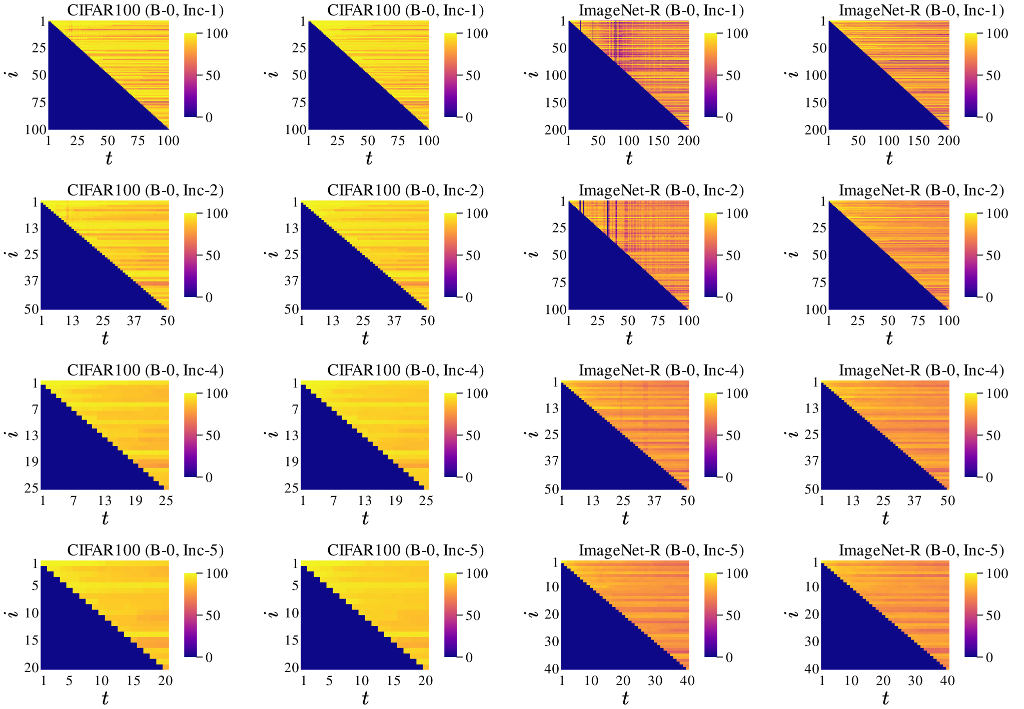
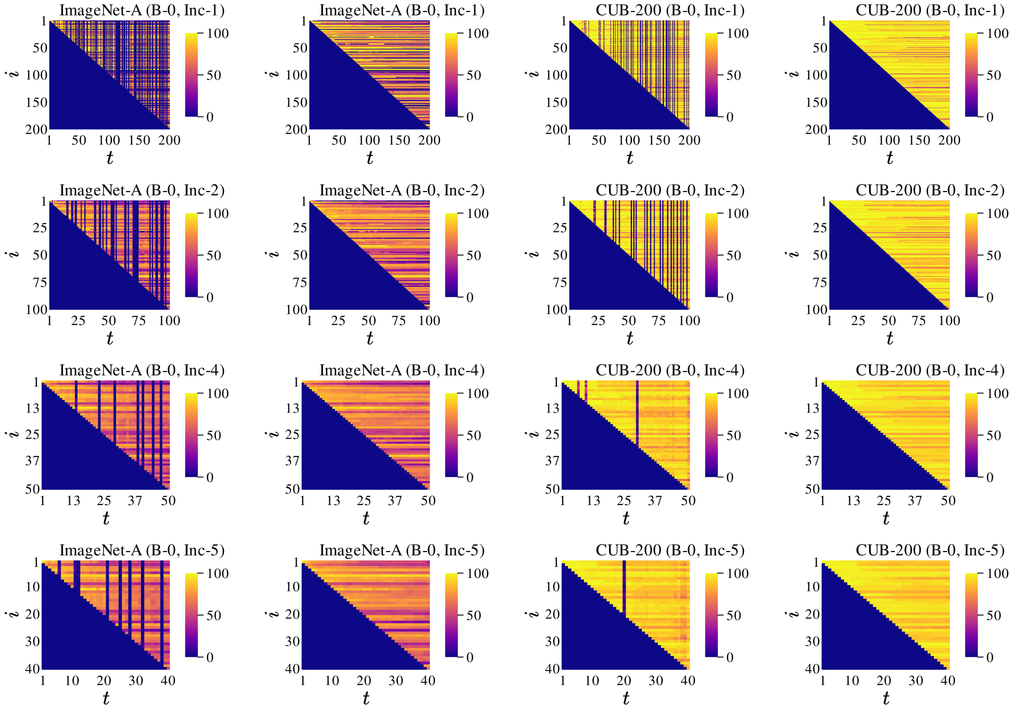


Figure 13: The average training MSE loss $\frac{1}{M_t}\|\boldsymbol{W}\boldsymbol{H}_{1:t} - \boldsymbol{Y}_{1:t}\|_{\mathrm{F}}^2$ of RanPAC for different ridge regularization parameters $\lambda \in \{0, 1, 10, 100, 1000\}$. Compare this with Fig. 2 and Fig. 3.



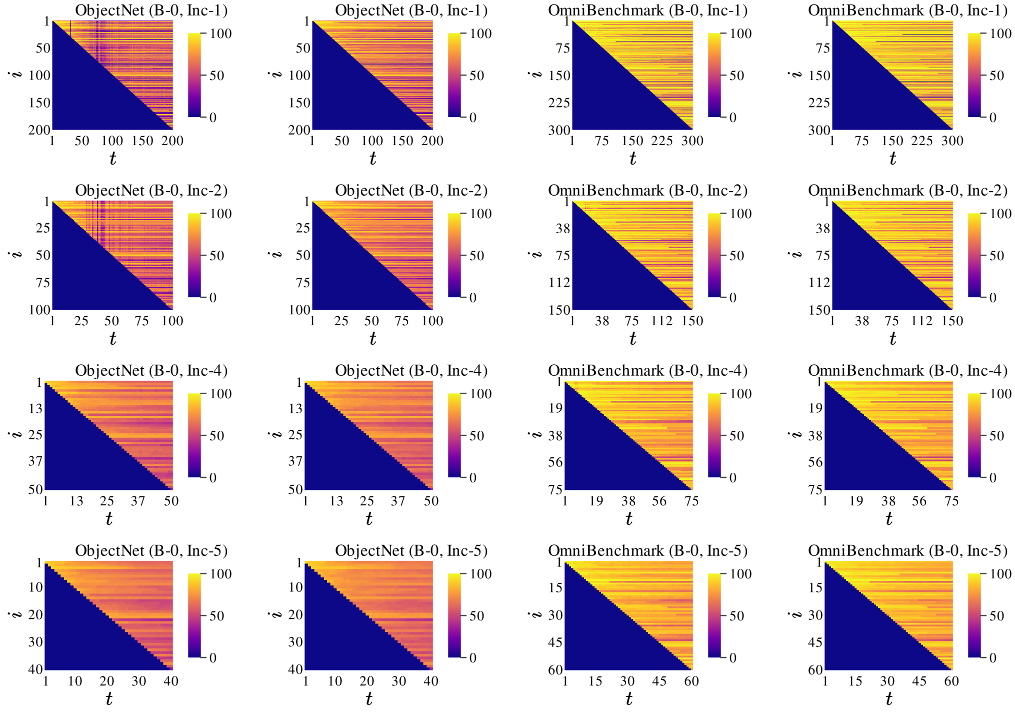
(14a) RanPAC (14b) ICL-TSVD (14c) RanPAC (14d) ICL-TSVD

Figure 14: Upper triangular accuracy matrices on CIFAR100 and ImageNet-R (Inc-1, 2, 4, 5).



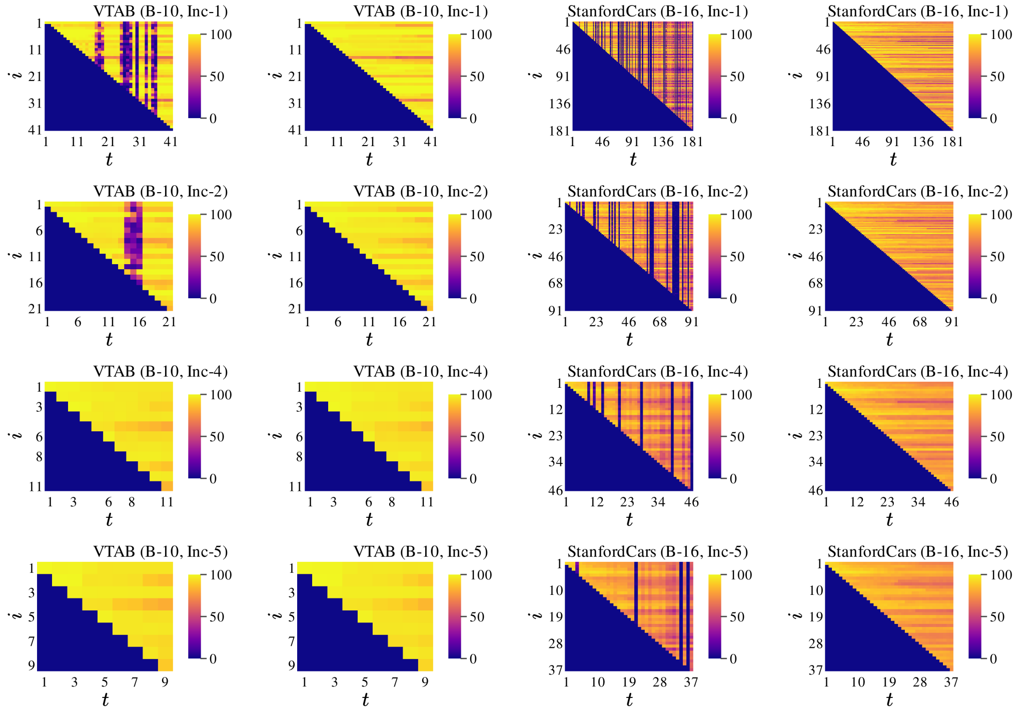
(15a) RanPAC (15b) ICL-TSVD (15c) RanPAC (15d) ICL-TSVD

Figure 15: Upper triangular accuracy matrices on ImageNet-A and CUB-200 (Inc-1, 2, 4, 5).



(16a) RanPAC (16b) ICL-TSVD (16c) RanPAC (16d) ICL-TSVD

Figure 16: Upper triangular accuracy matrices on ObjectNet and OmniBenchmark (Inc-1, 2, 4, 5).



(17a) RanPAC (17b) ICL-TSVD (17c) RanPAC (17d) ICL-TSVD

Figure 17: Upper triangular accuracy matrices on VTAB and StanfordCars (Inc-1, 2, 4, 5).

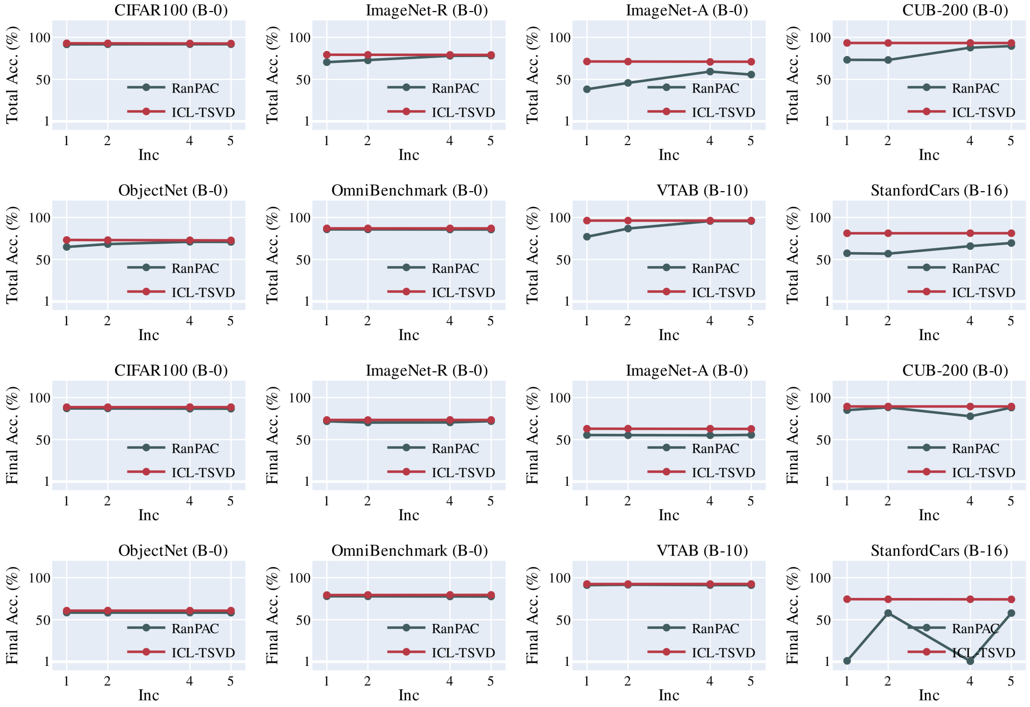


Figure 18: Our ICL-TSVD method is stable with respect to class increments of each task (Inc-1, 2, 4, 5), while the accuracy of RanPAC drops for smaller increments. The first two rows plots the total accuracy. Tthe last two rows plots the final accuracy. The figures here essentially plot the averages of the upper triangular accuracy matrices (total accuracy) or its last column (final accuracy) of Figs. 14 to 17. The numerical values of the total and final accuracy for Inc-1 are shown in Table 2.

L Extra Results for Rebuttal

L.1 Theory for the relation between the offline and online solutions (Reviewer iVyo)

Recall the definitions of the offline solution $\overline{\boldsymbol{W}}_t$ in (2) and the output $\widetilde{\boldsymbol{W}}_t$ of ICL-TSVD in (4):

$$\overline{\boldsymbol{W}}_t = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top \overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top, \quad \widetilde{\boldsymbol{W}}_t = \boldsymbol{Y}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top.$$

Reviewer iVyo had a question: Can we bound the distance $\|\overline{\boldsymbol{W}}_t - \widetilde{\boldsymbol{W}}_t\|_{\mathrm{F}}$?

To address this question, we consider the model $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t}$; this is the $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t} + \mathcal{E}_{1:t}$ with $\mathcal{E}_{1:t}$. Here we make this assumption for simplicity for purpose of the rebuttal, the result here can be extended to the case with noise, and we will include such a result in our next revision (even though the corresponding theorem will be much more complicated).

Recall the definition of gap_t in (17):

$$\mathrm{gap}_t := \mu_{k_t}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right) - \mu_{k_t+1}\left(\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top\right).$$

The following result addresses the question of **Reviewer iVyo** and it relies on Theorem 6.

Theorem 7. *Let a_t be defined as in (6) and gap_t as in (17). Assume $a_{t-1} < \left(1 - 1/\sqrt{2}\right)\mathrm{gap}_t$. Suppose $\boldsymbol{Y}_{1:t} = \boldsymbol{W}_t^* \boldsymbol{H}_{1:t}$. Then we have*

$$\left\|\overline{\boldsymbol{W}}_t - \widetilde{\boldsymbol{W}}_t\right\|_{\mathrm{F}} \leq \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left(\frac{\sqrt{2}a_{t-1}}{\mathrm{gap}_t} + \frac{t-1}{\gamma_t}\right). \tag{22}$$

Proof. Note that

$$\begin{aligned}
\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top &= \overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top \\
\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top &= \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top + \boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top,
\end{aligned}$$

where $\boldsymbol{D}_t$ is defined in (13) and the second equality follows from Lemma 1. So we have

$$\begin{aligned}
\left\|\overline{\boldsymbol{W}}_t - \widetilde{\boldsymbol{W}}_t\right\|_{\mathrm{F}} &= \left\|\boldsymbol{W}_t^* \boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{W}_t^* \boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\|_{\mathrm{F}} && (23) \\
&\leq \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left\|\boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{\Sigma}}_{1:t}^{-2}\overline{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{H}_{1:t}\boldsymbol{H}_{1:t}^\top \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\| && (24) \\
&= \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left\|\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top - \boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\| && (25) \\
&\leq \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left(\left\|\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\| + \left\|\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\|\right) && (26) \\
&\leq \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left(\left\|\overline{\boldsymbol{U}}_{1:t}\overline{\boldsymbol{U}}_{1:t}^\top - \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\| + \left\|\boldsymbol{D}_t \widetilde{\boldsymbol{U}}_{1:t}\widetilde{\boldsymbol{\Sigma}}_{1:t}^{-2}\widetilde{\boldsymbol{U}}_{1:t}^\top\right\|\right) && (27) \\
&\leq \|\boldsymbol{W}_t^*\|_{\mathrm{F}} \cdot \left(\frac{\sqrt{2}a_{t-1}}{\mathrm{gap}_t} + \frac{t-1}{\gamma_t}\right) && (28)
\end{aligned}$$

where the last inequality is due to Theorem 6 and Lemma 4. The proof is complete. □

L.2 Running times compared to other baselines (Reviewer iVyo)

Table 15: Running times (in minutes) of various methods on CIL datasets with $q_2 = 5$ (Inc-5). This is the same setting as Fig. 5.

	CIFAR100	ImageNet-R	ImageNet-A	CUB
L2P	52.09	72.31	26.96	17.62
CodaPrompt	174.9	246.94	27.28	39.91
EASE	139.74	128.35	54.0	66.23
ICL-TSVD ($E = 100$K)	31.24	27.0	1.29	3.73

L.3 Multiple runs of the experiments (Reviewer XHQX, Reviewer P9vj)

Table 16: Final and total accuracy of RanPAC and our method (ICL-TSVD) on CIL datasets with $q_2 = 1$ (Inc-1). Reported results are mean and standard deviations over 3 random seeds. See also Table 2 for similar results. In our next version, this table will replace Table 2.

	CIFAR100	ImageNet-R	ImageNet-A	CUB	ObjectNet	OmniBenchmark	VTAB	StanfordCars
				Final Accuracy				
RanPAC	86.99$_{\pm 0.06}$	**70.12**$_{\pm 0.39}$	**36.6**$_{\pm 25.35}$	**55.15**$_{\pm 37.14}$	57.14$_{\pm 0.24}$	77.9$_{\pm 0.04}$	91.47$_{\pm 0.3}$	**35.56**$_{\pm 24.75}$
Ours	88.19$_{\pm 0.05}$	**73.66**$_{\pm 0.07}$	**62.76**$_{\pm 0.16}$	**89.19**$_{\pm 0.06}$	60.82$_{\pm 0.15}$	79.3$_{\pm 0.06}$	92.51$_{\pm 0.05}$	**74.32**$_{\pm 0.11}$
				Total Accuracy				
RanPAC	90.46$_{\pm 0.73}$	**69.1**$_{\pm 0.37}$	**44.23**$_{\pm 0.46}$	**74.67**$_{\pm 2.87}$	**62.37**$_{\pm 2.1}$	85.23$_{\pm 0.56}$	**74.67**$_{\pm 3.07}$	**56.27**$_{\pm 0.78}$
Ours	92.18$_{\pm 0.56}$	**78.87**$_{\pm 0.34}$	**70.08**$_{\pm 0.86}$	**92.89**$_{\pm 0.59}$	**70.54**$_{\pm 1.94}$	86.51$_{\pm 0.59}$	**96.41**$_{\pm 0.31}$	**81.18**$_{\pm 0.68}$

Table 17: Final accuracy with pre-trained ViTs. Reported results are mean and standard deviations over 3 random seeds. See also Table 1. All methods of Table 1 will be included here in our next version. The table itself will stay in the appendix as the fonts are small and it will be very large.

(Part 1)	CIFAR100 (B-0)			ImageNet-R (B-0)			ImageNet-A (B-0)			CUB-200 (B-0)		
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20
RanPAC	86.93$_{\pm 0.15}$	87.06$_{\pm 0.04}$	87.02$_{\pm 0.06}$	71.95$_{\pm 0.13}$	71.85$_{\pm 0.06}$	72.29$_{\pm 0.23}$	**60.11**$_{\pm 3.03}$	61.07$_{\pm 1.4}$	61.71$_{\pm 0.32}$	88.07$_{\pm 0.33}$	87.53$_{\pm 0.33}$	87.83$_{\pm 0.54}$
Ours	88.21$_{\pm 0.02}$	88.23$_{\pm 0.03}$	88.24$_{\pm 0.04}$	73.62$_{\pm 0.1}$	73.6$_{\pm 0.13}$	73.67$_{\pm 0.02}$	**62.63**$_{\pm 0.08}$	62.96$_{\pm 0.19}$	62.89$_{\pm 0.23}$	89.2$_{\pm 0.11}$	89.16$_{\pm 0.16}$	89.14$_{\pm 0.18}$

(Part 2)	ObjectNet (B-0)			OmniBenchmark (B-0)			VTAB (B-10)			StanfordCars (B-16)		
	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20	Inc-5	Inc-10	Inc-20
RanPAC	58.15$_{\pm 0.17}$	58.17$_{\pm 0.33}$	58.21$_{\pm 0.43}$	77.74$_{\pm 0.09}$	77.48$_{\pm 0.1}$	77.45$_{\pm 0.2}$	91.53$_{\pm 0.14}$	91.77$_{\pm 0.08}$	91.77$_{\pm 0.13}$	**55.53**$_{\pm 3.65}$	**71.55**$_{\pm 0.11}$	**71.54**$_{\pm 0.12}$
Ours	60.71$_{\pm 0.09}$	60.67$_{\pm 0.16}$	60.61$_{\pm 0.12}$	79.63$_{\pm 0.09}$	79.65$_{\pm 0.04}$	79.76$_{\pm 0.04}$	92.47$_{\pm 0.01}$	92.56$_{\pm 0.06}$	92.53$_{\pm 0.04}$	**74.23**$_{\pm 0.11}$	**74.35**$_{\pm 0.13}$	**74.28**$_{\pm 0.09}$

L.4 Related Work in Continual Learning (**Reviewer XHQX**, **Reviewer P9vj**)

Reviewer XHQX and **Reviewer P9vj** suggested discussing a few methods that use online PCA in continual learning, including the PCA-OGD method of Doan et al. (2021). We fully agree with the comments. We will make the following structural changes to the main paper:

- In the new Section 2 (after the introduction), we will aim to describe our method directly, with some basic short motivations and setups. This amounts to merging Section 4 and part of Sections 2, 3 of the current manuscript; a lot of contents in Sections 2, 3 of the current manuscript will go to the appendix.
- In the subsequent section, we will discuss the connections to related methods. This includes continual learning methods visualized in Fig. 19. We will also discuss connections to classic concepts such as principal component regression (Remark 4) and incremental SVD (Appendix C). There, we will compare ICL-TSVD with RanPAC and its alternative implementations (Appendix C.3) in terms of time complexity and numerical robustness. We will also highlight GD versus SVD implementations for continual learning in the linear case (e.g., OGD versus Proposition 5 of ICL); we would argue that SVD-based implementations are more suitable for our case as the data is highly ill-conditioned, and we will try to verify this via experiments (we have seen that joint training of linear classifiers with the MSE loss is slower and less accurate than our method).
- Based on how the two new sections look, we will revise the introduction accordingly to make the paper consistent. We will begin to revise after recovering from the intense rebuttal period.

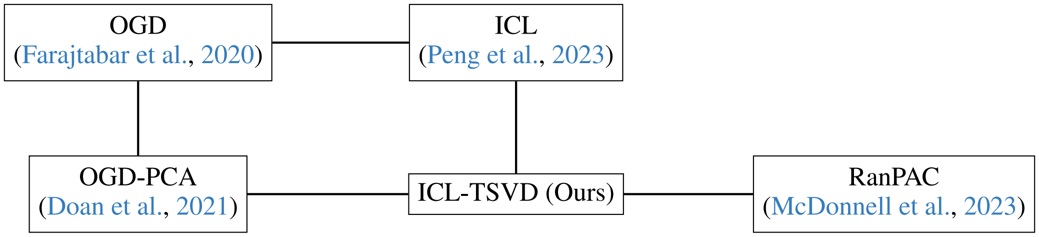


Figure 19: Relationship between several methods. The relation between OGD and ICL is described in Proposition 6 of Peng et al. (2023). PCA-OGD is inspired by OGD and adds SVD truncation on top of it. ICL-TSVD can be viewed as adding SVD truncation on top of ICL, and it borrows the network architecture from RanPAC (McDonnell et al., 2023). The differences between OGD-PCA and our method are explained in the rebuttal post. In the new section, we will have a concise discussion of all these and will also include a few follow-up works of OGD, including the method of Min et al. (2022).