Theory on Forgetting and Generalization of Continual Learning

Sen Lin [*1] **Peizhong Ju** [*1] **Yingbin Liang** [1] **Ness Shroff** [1 2]

Abstract

Continual learning (CL), which aims to learn a sequence of tasks, has attracted significant recent attention. However, most work has focused on the experimental performance of CL, and theoretical studies of CL are still limited. In particular, there is a lack of understanding on what factors are important and how they affect "catastrophic forgetting" and generalization performance. To fill this gap, our theoretical analysis, under overparameterized linear models, provides the first-known explicit form of the expected forgetting and generalization error for a general CL setup with an arbitrary number of tasks. Further analysis of such a key result yields a number of theoretical explanations about how overparameterization, task similarity, and task ordering affect both forgetting and generalization error of CL. More interestingly, by conducting experiments on real datasets using deep neural networks (DNNs), we show that some of these insights even go beyond the linear models and can be carried over to practical setups. In particular, we use concrete examples to show that our results not only explain some interesting empirical observations in recent studies, but also motivate better practical algorithm designs of CL.

1. Introduction

Continual learning (CL) (Parisi et al., 2019) is a learning paradigm where an agent needs to continuously learn a sequence of tasks. To resemble the extraordinary lifelong learning capability of human beings, the agent is expected to learn new tasks more easily based on accumulated knowledge from old tasks, and further improve the learning performance of old tasks by leveraging the knowledge of new tasks. The former is referred to as forward knowledge transfer and the latter as backward knowledge transfer. One major challenge herein is the so-called *catastrophic forgetting* (McCloskey & Cohen, 1989), i.e., the agent easily forgets the knowledge of old tasks when learning new tasks.

Although there have been significant efforts in experimental studies (e.g., Kirkpatrick et al. 2017; Chaudhry et al. 2018; Yoon et al. 2020; Doan et al. 2021; Evron et al. 2022) to address the forgetting issue, the theoretical understanding of CL is still in the early stage, where only a few attempts have emerged recently, e.g., Yin et al. 2020; Bennani et al. 2020; Doan et al. 2021; Evron et al. 2022 (see a more detailed discussion about the previous theoretical studies of CL in Section 2). However, none of these existing theoretical results provide an explicit characterization of forgetting and generalization error, that only depends on fundamental system parameters/setups (e.g., number of tasks/samples/parameters, noise level, task similarity/order). Thus, our work here provides the first-known explicit theoretical result in a more general CL setup with an arbitrary number of tasks, which enables us to comprehensively understand which factors are relevant and how they (precisely) affect forgetting and generalization error of CL.

Our main contributions can be summarized as follows.

First, we provide theoretical results on the expected value of forgetting and overall generalization error in CL, under a linear regression setup with *i.i.d.* Gaussian features and noise. The expression of our results is in an explicit form that captures a clear dependency on various system parameters/setups. Note that analyzing overparameterized linear models are important in their own right and also, as demonstrated in many recent works, are a first step towards understanding the generalization performance of DNNs, e.g., Belkin et al. 2018; Bartlett et al. 2020b; Ju et al. 2020; Muthukumar et al. 2020; Hastie et al. 2022.

Second, we investigate the impact of overparameterization, task similarity, and task ordering on both forgetting and generalization error of CL, which reveals the following important insights: 1) Both forgetting and generalization error can benefit from more parameters in the overparameterized regime. Moreover, benign overfitting exists and is easier to observe with large noise and/or low task similarity. 2) In terms of the impact of task similarity, we show that the generalization error always decreases when tasks become

[*]Equal contribution [1]Department of ECE, [2]Department of CSE, The Ohio State University, Columbus OH, USA. Correspondence to: Sen Lin <lin.4282@osu.edu>, Peizhong Ju <ju.171@osu.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

more similar, whereas this 'monotonicity' does not always hold for forgetting. Surprisingly, forgetting can even decrease when tasks are less similar under certain scenarios. 3) In order to minimize forgetting, the optimal task order should diversify the learning tasks in the early stage and learn more dissimilar tasks adjacently. This is also corroborated by some special scenarios where the tasks can be divided into multiple categories, and the optimal task order therein alternatively learns tasks from different categories.

Last but not least, we show that our findings for the linear models are applicable to and can also guide the algorithm designs for CL in practice, by conducting experiments on real datasets with DNNs. Specifically, our analysis of the impact of task similarity is clearly corroborated by the experimental results, which further sheds light on the recent observations (Ramasesh et al., 2020; Lee et al., 2021; Evron et al., 2022) that 'intermediate task similarity' leads to the worst forgetting in the two-task setup. Experimental results on the impact of task ordering are also consistent with our findings in linear models. More interestingly, inspired by our analysis of knowledge transfer in linear models, we slightly modify a previous method (Lin et al., 2022b) on leveraging task correlation to facilitate forward knowledge transfer, and show that better performance can be achieved by counting more on fresher old tasks. These encouraging results corroborate the benefits of studying the overparameterized linear models to fundamentally demystify CL.

2. Related Work

Empirical studies in CL. CL has attracted much attention in the past decade, and a vast amount of empirical methods have been proposed to address catastrophic forgetting. In general, the existing methods can be divided into three categories: (1) *Regularization-based methods* (e.g., Kirkpatrick et al. 2017; Aljundi et al. 2018; Liu & Liu 2022), which regularize the modifications on the important weights to old tasks when learning the new task; (2) *Parameter-isolation based methods* (e.g., Serra et al. 2018; Yoon et al. 2020; Yang et al. 2021), which learn a mask to fix the important weights to old tasks during the new task learning and further expand the neural network when needed; (3) *Memory-based methods*, which either store and replay data of old tasks when learning the new task, i.e., experience-replay based methods (e.g., Chaudhry et al. 2018; Riemer et al. 2018; Jin et al. 2021), or store the gradient information of old tasks and learn the new task in the orthogonal direction to old tasks without data replay, i.e., orthogonal-projection based methods (e.g., Farajtabar et al. 2020; Saha et al. 2021; Lin et al. 2022b).

Theoretical studies in CL. Specifically, (Bennani et al., 2020) and (Doan et al., 2021) analyzed generalization error and forgetting for the orthogonal gradient descent (OGD) approach (Farajtabar et al., 2020) based on NTK models, and further proposed variants of OGD to address forgetting. (Yin et al., 2020) proposed a unified framework for the performance analysis of regularization-based CL methods, by formulating them as a second-order Taylor approximation of the loss function for each task. (Asanuma et al., 2021) and (Lee et al., 2021) studied CL in the teacher-student setup to characterize the impact of task similarity on forgetting performance. (Cao et al., 2022) and (Li et al., 2022) investigated continual representation learning with dynamically expanding feature spaces, and developed provably efficient CL methods with a characterization of the sample complexity. (Chen et al., 2022) characterized the lower bound of memory in CL using the PAC framework. By investigating the information flow between neural network layers, (Andle & Yasaei Sekeh, 2022) analyzed the selection of frozen filters based on layer sensitivity to maximize the performance of CL. (Goldfarb & Hand, 2023) investigated the impact of overparameterization for linear models in a two-task setup. Nevertheless, none of these existing works show an explicit form of forgetting and generalization error, that only depends on fundamental system parameters/setups (e.g., number of tasks/samples/parameters, noise level, task similarity/order). In contrast, our work is the first one to provide such an explicit theoretical result in a more general CL setup with an arbitrary number of tasks, which enables us to comprehensively understand what factors (and how they) affect the forgetting and generalization performance of CL.

The most relevant study to our work is (Evron et al., 2022), which also studied CL in overparameterized linear models. However, our work is quite different from (Evron et al., 2022): (1) We study and provide the exact forms of both forgetting and generalization error based on the testing loss, while (Evron et al., 2022) only evaluated forgetting using the training data; (2) Our results characterize the performance of CL in a comprehensive way, through investigating how overparameterization, task similarity and task ordering affect both forgetting and generalization error, while (Evron et al., 2022) only studied the upper bound of catastrophic forgetting under specific task orderings; (3) Unlike (Evron et al., 2022), our study is able to explain recent phenomena and guide the algorithmic development in CL with DNN.

Studies about generalization performance on overparameterized models (benign overfitting). DNNs are usually so overparameterized that can completely fit all training samples, yet they can still generalize well on unseen test data. This seems to contradict the classical knowledge of *bias-variance trade-off*. As a first step of understanding this mystery, the "benign overfitting" or "double-descent" phenomenon[1] has been discovered and studied for overfit-

[1] i.e., test error decreases again in the overparameterized re-

ted solutions of single-task linear regression. For example, some work discovered and studied double-descent with min ℓ_2-norm overfitted solutions (Belkin et al., 2018; 2019; Bartlett et al., 2020a; Hastie et al., 2019; Muthukumar et al., 2019) or min ℓ_1-norm overfitted solutions (Mitra, 2019; Ju et al., 2020), while using simple features such as Gaussian or Fourier features. Some other recent work studied the overfitted generalization performance by adopting features that approximate shallow neural networks, for example, random feature (RF) models (Mei & Montanari, 2019), two-layer neural tangent kernel (NTK) models (Arora et al., 2019; Satpathi & Srikant, 2021; Ju et al., 2021), and three-layer NTK models (Ju et al., 2022). All of these studies considered only a single task. In contrast, our work focuses on CL with a sequence of tasks, which brings in many new variables such as task similarity and task ordering.

3. Continual Learning in Linear Models

Consider the standard CL setup where a sequence of tasks $\mathbb{T} = \{1, ..., T\}$ arrives sequentially in time.

Ground truth. We consider a linear ground truth (Belkin et al., 2018; Evron et al., 2022) for each task. Specifically, for task t, the output $y \in \mathbb{R}$ is given by

$$y_t = \hat{\boldsymbol{x}}_t^\top \hat{\boldsymbol{w}}_t^* + z_t, \tag{1}$$

where $\hat{\boldsymbol{x}}_t \in \mathbb{R}^{s_t}$ denotes the feature vector, $\hat{\boldsymbol{w}}_t^* \in \mathbb{R}^{s_t}$ denotes the model parameters, and z_t is the random noise. Here s_t denotes the number of features of ground truth (i.e., the number of true features). In practice, true features are unknown in advance. Therefore, when choosing a model to learn a certain task, people usually choose more features than enough such that all possible features are included. We write this formally into the following assumption[2].

Assumption 3.1. We index all possible features by $1, 2, \cdots$. Let $\mathcal{W}$ denote the set of indices of all the chosen features in the model to be trained, with cardinality $|\mathcal{W}| = p$. Let $\mathcal{S}_t$ denote the set of indices of t-th task's true features, with cardinality $|\mathcal{S}_t| = s_t$. We assume that $\bigcup_{t\in\mathbb{T}} \mathcal{S}_t \subseteq \mathcal{W}$.

We next define an expanded ground-truth vector $\boldsymbol{w}_t^* \in \mathbb{R}^p$ that expands the original ground-truth vector $\hat{\boldsymbol{w}}_t^*$ from dimension s_t to dimension p by filling zeros in the positions $\mathcal{W} \setminus \mathcal{S}_t$. Let $\boldsymbol{x}_t$ be the corresponding features for $\boldsymbol{w}_t^*$. Therefore, the ground truth Equation (1) can be rewritten as

$$y_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t^* + z_t. \tag{2}$$

Data. For each task $t \in \mathbb{T}$, the training dataset is denoted as $\mathcal{D}_t = \{(\boldsymbol{x}_{t,j}, y_{t,j}) \in \mathbb{R}^p \times \mathbb{R}\}_{j\in[n_t]}$ with sample size n_t. By stacking the training data as $\boldsymbol{X}_t \coloneqq [\boldsymbol{x}_{t,1}\ \boldsymbol{x}_{t,2}\ \cdots\ \boldsymbol{x}_{t,n_t}] \in \mathbb{R}^{p\times n_t}$ and $\boldsymbol{y}_t \coloneqq [y_{t,1}\ y_{t,j}\ \cdots\ y_{t,n_t}]^\top \in \mathbb{R}^{n_t\times 1}$, Equation (2) can be written as $\boldsymbol{y}_t = \boldsymbol{X}_t^\top \boldsymbol{w}_t^* + \boldsymbol{z}_t$.

To simplify our analysis, we consider *i.i.d.* Gaussian features and noise, which is stated in the following assumption.

Assumption 3.2. Each element of $\boldsymbol{X}_t$ for all $t \in \mathbb{T}$ follows standard Gaussian distribution $\mathcal{N}(0, 1)$ and is independent of each other. The noise $\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma_t^2 \boldsymbol{I}_p)$ and is independent of each other for all $t \in \mathbb{T}$, where $\sigma_t \geq 0$ denotes the noise level.

Learning procedure. We train the model parameters $\boldsymbol{w}$ for each task sequentially. Let $\boldsymbol{w}_t$ denote the result after training for task t, which is also the initial point in the model training for task $t+1$. Let $\boldsymbol{w}_0 = \boldsymbol{0}$, i.e., task 1 starts training from zero. For each task t, the training loss is defined by mean-squared-error (MSE) with respect to (w.r.t.) $(\boldsymbol{X}_t, \boldsymbol{y}_t)$:

$$\mathcal{L}_t^{tr}(\boldsymbol{w}, \mathcal{D}_t) = \tfrac{1}{n_t}\|(\boldsymbol{X}_t)^\top \boldsymbol{w} - \boldsymbol{y}_t\|_2^2. \tag{3}$$

When underparameterized (i.e., $n_t \leq p$), minimizing Equation (3) has a unique solution (with probability 1). When overparameterized (i.e., $p > n_t$), minimizing Equation (3) has an infinite number of solutions that make Equation (3) zero. Among all overfitted solutions, we are particularly interested in the one corresponding to the convergent point of stochastic gradient descent (SGD) for minimizing Equation (3). In fact, it can be shown that such an overfitted solution has the smallest ℓ_2-norm of the change of parameters (Gunasekar et al., 2018). In other words, $\boldsymbol{w}_t$ corresponds to the solution to the following optimization problem:

$$\min_{\boldsymbol{w}}\ \|\boldsymbol{w} - \boldsymbol{w}_{t-1}\|_2, \quad s.t.\ (\boldsymbol{X}_t)^\top \boldsymbol{w} = \boldsymbol{y}_t. \tag{4}$$

The constraint in Equation (4) implies that the training loss is exactly zero (i.e., overfitted).

Performance evaluation. For the described linear system, we use $\mathcal{L}_t(\boldsymbol{w})$ to denote the model error[3] for task t:

$$\mathcal{L}_t(\boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{w}_t^*\|^2, \tag{5}$$

which characterizes the generalization performance of $\boldsymbol{w}$ on task t. As is standard in the empirical studies of CL, e.g., (Chaudhry et al., 2018; Lin et al., 2022b), we evaluate the performance of CL on two key metrics, forgetting and overall generalization error, defined as below:

(1) *Forgetting:* It measures how much 'knowledge' of old tasks has been forgotten after learning the current task. Specifically, after learning task $t \in [2, T]$, the average forgetting over all old tasks $i \in [1, t-1]$ is defined as:

$$F_t = \tfrac{1}{t-1}\textstyle\sum_{i=1}^{t-1}(\mathcal{L}_i(\boldsymbol{w}_t) - \mathcal{L}_i(\boldsymbol{w}_i)). \tag{6}$$

In Equation (6), $\mathcal{L}_i(\boldsymbol{w}_t) - \mathcal{L}_i(\boldsymbol{w}_i)$ denotes the performance difference between $\boldsymbol{w}_i$ (the result after training task i) and $\boldsymbol{w}_t$ (the result after training task t) on test data of task i.

(2) *Overall generalization error:* We evaluate the model

gion with more parameters, so the overfitting is benign for the generalization performance.

[2]When Assumption 3.1 does not hold, the derivation techniques for Theorem 4.1 in the next section still hold with a minor modification that treats the missing features as noise.

[3]It can be proved that the model error we defined here is equivalent to the mean-squared-error on noise-free test data.

generalization performance of the final task model $\boldsymbol{w}_T$ in terms of the average model error over all tasks:

$$G_T = \frac{1}{T}\sum_{i=1}^{T}\mathcal{L}_i(\boldsymbol{w}_T). \tag{7}$$

It is worth noting that the forgetting defined in (Evron et al., 2022) is based on the training loss, which consequently ignores the generalization performance of the learned models for old tasks. Such a definition is not only inconsistent with the evaluation metric in empirical studies, but also insufficient to capture the backward knowledge transfer because the value of forgetting therein can not be negative.

We further simplify the current setup by assuming that each task has the same number of training samples as well as the same noise level σ, stated as follows.

Assumption 3.3. $n_t = n$ and $\sigma_t = \sigma$ for all $t \in \mathbb{T}$.

Note that Assumption 3.3 is adopted only to make our results (which will be shown in the next section) easy to interpret. In fact, our analysis can be easily generalized to the situation when Assumption 3.3 does not hold.

4. Main Results and Interpretations

Although we use linear models, in order to provide hints on understanding DNNs that are usually heavily overparameterized, we are particularly interested in the performance of CL in the overparameterized region ($p > n$), where we define the overparameterized ratio as $r := 1 - \frac{n}{p}$. For ease of exposition, we define the following coefficients that will appear in our main theorem:

$$c_{i,j} := (1-r)\left(r^{T-i} - r^{j-i} + r^{T-j}\right), \tag{8}$$

where $1 \le i < j \le T$ are the indices of tasks. Now we are ready to state our main theorem that characterizes the expected value of forgetting and overall generalization error:

Theorem 4.1. *When $p \ge n+2$, we must have*

$$\mathbb{E}[F_T] = \frac{1}{T-1}\sum_{i=1}^{T-1}\Big[\underbrace{(r^T - r^i)\|\boldsymbol{w}_i^*\|^2}_{\text{Term F1}} + \underbrace{\sum_{j>i}^{T} c_{i,j}\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2}_{\text{Term F2}} + \underbrace{\frac{p\sigma^2}{p-n-1}(r^i - r^T)}_{\text{Term F3}}\Big] \tag{9}$$

$$\mathbb{E}[G_T] = \underbrace{\frac{r^T}{T}\sum_{i=1}^{T}\|\boldsymbol{w}_i^*\|^2}_{\text{Term G1}} + \underbrace{\frac{1}{T}\sum_{i=1}^{T}\frac{nr^{T-i}}{p}\sum_{k=1}^{T}\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2}_{\text{Term G2}} + \underbrace{\frac{p\sigma^2}{p-n-1}\left(1 - r^T\right)}_{\text{Term G3}}. \tag{10}$$

To the best of our knowledge, Theorem 4.1 is the first result that establishes the closed forms of forgetting and overall generalization error of CL in overparameterized linear models. In the rest of the paper, we will see that Theorem 4.1 not only describes how CL performs on the linear system but also provides guidance on applying CL in practice that DNNs and real-world datasets. The proof of Theorem 4.1 is in Appendix D.3. We also verify the correctness of Theorem 4.1 in Figure 1 where discrete points indicated by markers in Figure 1 (drawn by simulations) are very close to the curves (drawn by Theorem 4.1 and Theorem 4.3).

We can further simply Equation (9) and Equation (10) by only considering two tasks, so as to better understand Theorem 4.1. The result is shown in the following corollary, which clearly characterizes the dependence on task similarity and different system parameters.

Corollary 4.2. *When $T = 2$ and $p \ge n+2$, we must have*

$$\mathbb{E}[F_2] = (r^2 - r)\|\boldsymbol{w}_1^*\|^2 + \frac{n}{p}\|\boldsymbol{w}_2^* - \boldsymbol{w}_1^*\|^2 + \frac{nr\sigma^2}{p-n-1}, \tag{11}$$

$$\mathbb{E}[G_2] = \frac{r^2}{2}\left(\|\boldsymbol{w}_1^*\|^2 + \|\boldsymbol{w}_2^*\|^2\right) + \frac{1-r^2}{2}\|\boldsymbol{w}_1^* - \boldsymbol{w}_2^*\|^2 + \frac{p\sigma^2(1-r^2)}{p-n-1}. \tag{12}$$

Based on Theorem 4.1, we will provide insights on the following three aspects.

(1) *Overparameterization (Section 4.1).* In order to understand the generalization power of overfitted machine learning models, much attention has focused (e.g., Belkin et al. 2018; Ju et al. 2020; Hastie et al. 2022) on studying the impact of overparameterization on single-task learning, whereas how overparameterization affects the performance of CL still remains unclear. Fortunately, the exact forms in Theorem 4.1 provide a way to directly evaluate the impact of overparameterization and the random noise on both forgetting and generalization error in CL.

(2) *Task similarity (Section 4.2).* Both forgetting and generalization error depend on the optimal model gap between any two tasks , i.e., $\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2$ for any task k and i, which defines the task similarity in this work (smaller gap means higher similarity). This model gap can be treated as the characterization of the task similarity, which depends only on the underlying true model of each task and is independent of the learnt models. Understanding the impact of task similarity is helpful to not only explain empirical observations but also guide better designs of CL in practice.

(3) *Task ordering (Section 4.3).* Given a fixed set of tasks in CL, the learning order of the task sequence clearly plays an important role in affecting both $\mathbb{E}[F_T]$ and $\mathbb{E}[G_T]$, through the task order-dependent coefficients, e.g., c_{ij} in Equation (9) and r^{T-i} in Equation (10). For example, suppose $\|\boldsymbol{w}_i^*\|^2$ is the same for all $i \in \mathbb{T}$, the optimal task ordering to minimize the generalization error is to learn the tasks in a decreasing order of $\sum_{k=1}^{T}\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2$, i.e., $i < j$ if $\sum_{k=1}^{T}\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2 > \sum_{k=1}^{T}\|\boldsymbol{w}_k^* - \boldsymbol{w}_j^*\|^2$. Intuitively, the most dissimilar task should be learnt first in this case. Inves-

tigating the impact of task ordering is particularly valuable when the agent can control the task order in CL, in the same spirit of curriculum learning (Bengio et al., 2009).

In what follows, we will delve into the impact of those three crucial factors in order to provide a comprehensive understanding of CL in the linear models.

4.1. The impact of overparameterization

In this subsection, we show some insights about the impact of overparameterization. Specially, we will discuss what happens when p changes under a fixed n.

1) More parameters can lead to zero forgetting and alleviate the negative impact of task dissimilarity on generalization error. As shown in Theorem 4.1, when $p \to \infty$, we can have that $\mathbb{E}[F_T] \to 0$ and Term G2 also approaches zero. In some special cases, we can further show that Term G2 is monotonically decreasing w.r.t. p. A more detailed discussion can be found in Appendix C.3.

2) Benign overfitting exists and is easier to observe with large noise and/or low task similarity. As we introduced in related work, benign overfitting has recently been discovered and studied in linear models as a first step towards understanding why DNNs can still generalize well even when heavily overparameterized. The concept of "benign overfitting" and "double-descent" is initially proposed for only a single task. We now show that such a phenomenon also exists in CL where there exists a sequence of tasks.

Notice that Theorem 4.1 is for the overparameterized region. For a precise comparison between the performance of overfitting and underfitting, we present the theoretical result of the underparameterized region in the following theorem.

Theorem 4.3. *When $n \geq p+2$, we must have*

$$\mathbb{E}[F_T] = \frac{1}{T-1}\sum_{i=1}^{T-1}\|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2,$$

$$\mathbb{E}[G_T] = \left(\frac{1}{T}\sum_{i=1}^{T-1}\|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2\right) + \frac{p\sigma^2}{n-p-1}.$$

We provide an intuitive explanation and rigorous proof of Theorem 4.3 in Appendix D.8. As shown in Theorem 4.3, $\mathbb{E}[G_T]$ becomes larger when the noise level σ is larger, and both $\mathbb{E}[F_T]$ and $\mathbb{E}[G_T]$ become larger when tasks are less similar (i.e., when $\sum_{i=1}^{T-1}\|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2$ is larger). In contrast, in the overfitted situation, Term F2 and Term G2 in Theorem 4.1 (corresponding to task similarity), Term F3 and Term G3 (corresponding to noise) will go to zero when $p \to \infty$. This indicates that when the noise level is high and/or task similarity is low, the performance of CL in the overparameterized situation is more likely to be better than that in the underparameterized situation, i.e., benign overfitting exists and is easier to observe. This can be observed from Figure 1. For example, the blue curve with markers "+" corresponds to the largest noise (compared with other curves in Figure 1(d)) and the lowest task similarity (compared with Figure 1(c)), and it has the deepest descent curve in the overparameterized region ($p > 50 = n$). This observation indicates that benign overfitting is easier to observe with larger noise and lower task similarity.

3) A descent floor sometimes exists on forgetting and generalization error, especially when tasks are similar and noise is low. In Equation (11), the term $(r^2 - r)\|\boldsymbol{w}_1^*\|^2$ first decreases and then increases as p increases from n to ∞ (i.e., r increases from 0 to 1), while the remaining two terms decrease as p increases. Thus, when $\|\boldsymbol{w}_2^* - \boldsymbol{w}_1^*\|^2$ (task similarity) and σ^2 (noise level) are relatively small, the trend of F_2 w.r.t. p will be dominated by the first term, where a descent floor of forgetting exists. In the right-hand-side of Equation (12), the first term increases as p increases, while the rest two terms decrease as p increases. Taking the derivative of Equation (12) on p, we have

$$\frac{\partial \mathbb{E}[G_2]}{\partial p} = \frac{2nr\boldsymbol{w}_1^{*\top}\boldsymbol{w}_2^*}{p^2} - \sigma^2\left(\frac{(n+1)(1-r^2)}{(p-n-1)^2} + \frac{2nr}{(p-n-1)p}\right).$$

Here, since $\frac{1}{p-n-1}$ is very large when p is close to n, while decreasing to zero when $p \to \infty$, we can tell that when σ^2 is relatively small w.r.t. $\boldsymbol{w}_1^{*\top}\boldsymbol{w}_2^*$, $\frac{\partial \mathbb{E}[G_2]}{\partial p}$ will be positive and then negative as p increases from $n+2$ to ∞. In other words, if these two tasks have a positive correlation (i.e., $\boldsymbol{w}_1^{*\top}\boldsymbol{w}_2^* > 0$) and noise is small, there exists a descent floor w.r.t. p on $\mathbb{E}[G_2]$. Such a phenomenon can exist in other setups besides the special case of $T = 2$. For example, in Figure 1(a)(c) where the ground truth for each task is exactly the same, we can observe a descent floor for the small noise cases $\sigma = 0.3$ and 0.1 (i.e., orange and green curves with markers "×" and "Y", respectively).

4.2. The impact of task similarity

Generalization error monotonically decreases with task similarity whereas forgetting may not. Based on Theorem 4.1, it can be seen that the generalization error $G_T(\boldsymbol{w}_T)$ decreases when $\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2$ for any two different tasks k and i decreases, because of the positive coefficients in Term G2 in Equation (10). Intuitively, the generalization error of CL will be smaller if the tasks are more similar with each other. In contrast, the forgetting F_T may not change monotonically with $\|\boldsymbol{w}_k^* - \boldsymbol{w}_i^*\|^2$, because the coefficients c_{ij} in Term F2 in Equation (9) can be negative. To verify this result, we consider two different scenarios.

(1) Consider the case where $T = 2$. In Equation (11), $\|\boldsymbol{w}_2^* - \boldsymbol{w}_1^*\|^2$ captures the task similarity between tasks 1 and 2 in terms of the optimal task models. It is clear that forgetting increases with $\|\boldsymbol{w}_2^* - \boldsymbol{w}_1^*\|^2$, i.e., less forgetting when the two tasks are more similar.

(2) Consider the case where $T = 4$. We first assume that $\|\boldsymbol{w}_i^*\|^2 = w$ for any task $i \in [1, 4]$ considering the overpa-

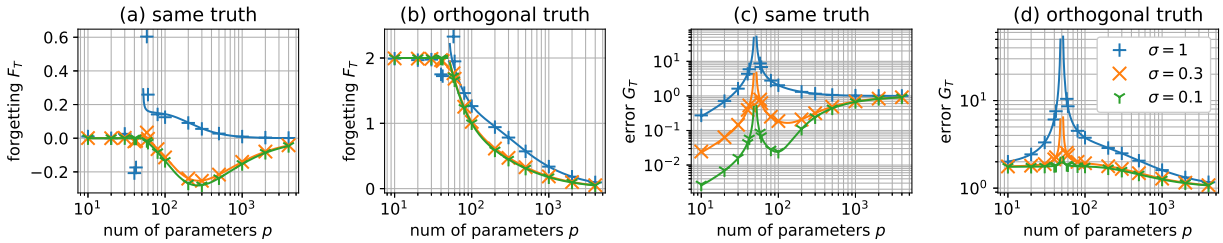


Figure 1. The trend of forgetting and overall generalization error w.r.t. the number of model parameters, where $T = 8$, $n = 50$, $\hat{\boldsymbol{w}}_t^* \in \mathbb{R}^{10}$ and $\|\hat{\boldsymbol{w}}_t^*\|^2 = 1$ for all $t \in \mathbb{T}$. The ground truths are the same for all tasks in Subfigures (a) and (c), but are orthogonal in Subfigures (b) and (d) where $\hat{\boldsymbol{w}}_t^*$ equals to t-th standard basis for all $t \in \mathbb{T}$. Here the orthogonal truth means that for two tasks i and j, they do not have common features, i.e., $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$. In this case, the inner product of their ground truth model is equal to zero. The discrete points indicated by markers are calculated by simulation and are the average of 300 random simulation runs. The curves are drawn by the theoretical expressions in Theorem 4.1 and Theorem 4.3.

rameterized models (Evron et al., 2022). Suppose that task 1 and task 2 share the same set of true features, which is orthogonal to the feature set of both task 3 and task 4, i.e., $\mathcal{S}_1 = \mathcal{S}_2$ and $\mathcal{S}_1 \cap (\mathcal{S}_3 \cup \mathcal{S}_4) = \emptyset$. Note that

$$\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2 = \|\boldsymbol{w}_i^*\|^2 + \|\boldsymbol{w}_j^*\|^2 - 2\langle \boldsymbol{w}_i^*, \boldsymbol{w}_j^* \rangle$$

where $\langle \boldsymbol{w}_i^*, \boldsymbol{w}_j^* \rangle = 0$ if $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$. Therefore, we can control the value of $\|\boldsymbol{w}_1^* - \boldsymbol{w}_2^*\|^2$ by changing $\boldsymbol{w}_2^*$, without affecting the value of $\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$ for any pair of $\{i, j\} \neq \{1, 2\}$. Based on Theorem 4.1, it can be shown that $c_{1,2} < 0$, such that increasing $\|\boldsymbol{w}_1^* - \boldsymbol{w}_2^*\|^2$, i.e., the tasks become less similar, will surprisingly decrease forgetting.

4.3. The impact of task ordering

In order to investigate the impact of task ordering on the performance of CL, we assume that $\|\boldsymbol{w}_t^*\|^2 = w$ for every task $t \in \mathbb{T}$. By ignoring the task order-independent terms in Equation (9) and Equation (10), we focus on the task order-dependent terms, i.e., Term F2 and Term G2.

1) Optimal task ordering of minimizing forgetting tends to arrange dissimilar tasks adjacently in the early stage of the sequence. As shown in Term F2, the optimal task order to minimize forgetting closely hinges upon the value of $c_{i,j}$. Based on Equation (8), $c_{i,j}$ is smaller when (1) i and j are smaller and (2) they are closer. Intuitively, this implies that tasks with larger $\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$ should be learnt adjacently with higher priority in CL, in order to minimize the impact of the task dissimilarity on the value of $\tilde{F}_T(\boldsymbol{w}_T)$. However, finding the optimal task order for the general case is highly nontrivial due to the complex coupling across $\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$ for different tasks. To verify the implication above and better understand the structure of the optimal task order, we study several special cases of the task setups.

(1) *[Special case I: One vs Many]* There are two different categories of tasks, where tasks in the same category have the same optimal model; among the entire task set, one special task belongs to Category I while the other tasks belong to Category II. In this case, the optimal task order is captured by the optimal learning order of the special task in Category I. We have the following result to characterize the optimal task order for Special case I.

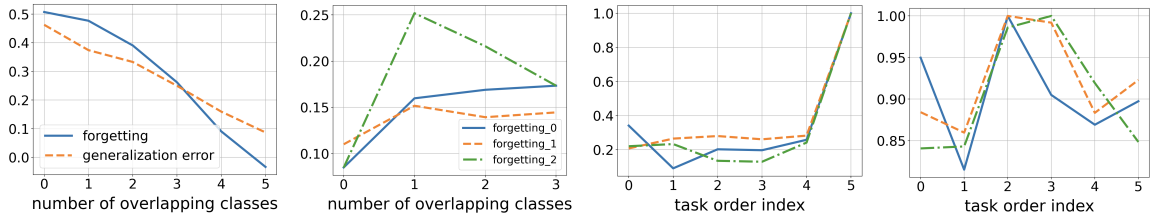
Proposition 4.4. *Let $i^* \in [1, T]$ denote the optimal order of the special task in Category I to minimize forgetting. Suppose $p \geq n + 2$. Then 1) i^* can take any integer value between 2 and $\frac{T}{2}$, depending on the value of $\frac{n}{p}$; 2) i^* is non-decreasing with $\frac{n}{p}$.*

As indicated by Proposition 4.4, the special task will be learnt in the first half of the sequence, such that the task diversity in the first half is always larger than in the second half. Besides, with the model capacity increasing ($\frac{n}{p} \to 0$), the order of the special task will move towards the beginning of the sequence, because 1) the model is less concerned about the special task since it is powerful enough to learn different features and 2) the model focuses on the performance of the majority and seeks to learn more tasks from Category II at the end of the sequence for better performance.

(2) *[Special case II: Equal Occurrence]* There are two different categories (C_1 and C_2) of tasks, where tasks in the same category have the same optimal model; particularly, two categories contain the same number of tasks. If task $1 \in C_1$ and task $2 \in C_2$, we will denote the task order as (C_1, C_2). The following proposition characterizes the optimal task order in this case:

Proposition 4.5. *Suppose $p \geq n + 2$. For $T = 4$ and $T = 6$, the optimal task order to minimize forgetting is the perfectly alternating order, i.e., (C_i, C_j, C_i, C_j) and $(C_i, C_j, C_i, C_j, C_i, C_j)$, where $i, j \in \{1, 2\}$ and $i \neq j$.*

Proposition 4.5 clearly shows that adjacent tasks always belong to different categories in the optimal task order, which leads to a more diverse task learning sequence. Intuitively, the alternating order maximizes the memorization of each category by keeping practicing on different tasks. It can be further proved that the perfectly alternating order is also optimal for $T = 6$ with three different categories (Appendix C.4). Based on these results, we expect that such an alternating order may minimize forgetting for more general scenarios where the tasks contain multiple categories with



(a) (b) (c) (d)

Figure 2. Impact of task similarity and task order. (a) When $T = 2$, both forgetting and generalization error decrease when two tasks have more overlapping classes. (b) When $T = 4$, the forgetting surprisingly increases when the first two tasks have more overlapping classes; 'forgetting_0', 'forgetting_1' and 'forgetting_2' correspond to three cases of the task setups (also in (c) and (d)). (c) Consider one special task and five same tasks in CL with $T = 6$; the task order index shows the order of the special task, and the smallest forgetting is achieved always when the special task is learnt in the first half for each case (we normalize the forgetting w.r.t. the worst forgetting in each case). (d) Consider two categories of tasks in CL with $T = 4$; the task order indices 0 and 1 refer to the perfectly alternating orders, one of which always achieve the smallest forgetting among all possible orders. All the results are averaged over 3 random seeds.

equal cross-category task model distance.

The findings on the optimal task order indeed share similar insights with the surprising impact of task correlation on forgetting mentioned earlier. Intuitively, learning more dissimilar tasks in the early stage facilitates the exploration of a larger feature space and expands the learnt feature space in CL, which can make the learning of similar tasks in the future much easier. In the meanwhile, the impact of task similarity among the early tasks continuously diminishes in CL with T increasing, as suggested by the coefficients $c_{i,j}$ (which can be smaller for smaller i, j) in Theorem 4.1. Therefore, the negative impact of learning more dissimilar tasks on forgetting is weaker when they are learnt in the early stage, compared to being learnt in the late stage.

2) The optimal task ordering for minimizing forgetting and for minimizing generalization error are not always the same. Consider Special case I and Special case II. It can be shown that the optimal task orders for minimizing forgetting and generalization error are different in Special case I but same in Special case II. This would open up an interesting direction of finding the task order with balanced impact on forgetting and generalization error. A more detailed discussion can be found in Appendix C.4.

5. Implications on CL with DNN

So far, we have explored different aspects that affect the performance of CL in overparameterized linear models. More interestingly, we will show next that Theorem 4.1 can also shed light on CL in practice with DNNs, by reflecting on recent empirical observations and guiding improved designs therein. More experimental details are in Appendix A.

5.1. Forgetting is not always monotonic with task similarity

To see if our understandings about the impact of task similarity on forgetting can be carried over to CL with DNN, we conduct experiments on MNIST (LeCun et al., 1989) using a convolutional neural network to investigate the impact of task similarity therein. More specifically, we consider each task i as a binary classification problem which seeks to decide if an image belongs to a task-specific label subset Y_i of the classes, i.e., $Y_i \subset \{0, ..., 9\}$ in MNIST, and we control the task similarity through the degree of class overlapping between the task-specific subsets, e.g., task i and j are more similar if the cardinality of $Y_i \cap Y_j$ is larger.

We first consider the case with two tasks, where we fix Y_1 for task 1 as $\{0, 1, 2, 3, 4\}$ and change Y_2 for task 2 to have different numbers of overlapping classes with Y_1. As shown in Figure 2(a), both forgetting and generalization error decrease when the number of overlapping classes increases, i.e., the two tasks are more similar, which is indeed consistent with our analysis for the overparameterized linear models for $T = 2$. More interestingly, this result also agrees with some recent studies (Ramasesh et al., 2020; Lee et al., 2021; Evron et al., 2022), which found that 'intermediate task similarity' leads to the worst forgetting in a two-task setup using various notions of task similarity (different from our definition of task similarity using the optimal model gap), through either empirical studies or analyzing the upper bound of forgetting. We can build the connection based on the closed form of forgetting F_2 in Equation (11).

Note that in Equation (11)

$$\|\boldsymbol{w}_2^* - \boldsymbol{w}_1^*\|^2 = \|\boldsymbol{w}_2^*\|^2 + \|\boldsymbol{w}_1^*\|^2 - 2\langle \boldsymbol{w}_1^*, \boldsymbol{w}_2^* \rangle$$

and we can divide the task correlation into three cases depending on the value of $\langle \boldsymbol{w}_1^*, \boldsymbol{w}_2^* \rangle$: (1) $\langle \boldsymbol{w}_1^*, \boldsymbol{w}_2^* \rangle = 0$: Two tasks are orthogonal in the sense that they share no common features, i.e., $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$; (2) $\langle \boldsymbol{w}_1^*, \boldsymbol{w}_2^* \rangle > 0$: Two tasks share some common features and are 'positively' correlated; (3) $\langle \boldsymbol{w}_1^*, \boldsymbol{w}_2^* \rangle < 0$: Two tasks share some common features but are 'negatively' correlated. Compared to the first case when two tasks are orthogonal, it can be easily shown that forgetting is worse when two tasks are negatively correlated even if they share some common features, which indeed corresponds to 'the intermediate task similarity' in (Ramasesh et al., 2020; Lee et al., 2021; Evron et al., 2022).

The reason behind is that in this case task 2 updates the model in the opposite direction to the model update of task 1, which inevitably leads to more forgetting in CL. Note that in Figure 2(a), the non-overlapping case means that task 1 and 2 are negatively correlated because in this two-task case the image that is not in Y_1 must be in Y_2. On the other hand, the forgetting can even be negative when the two tasks are positively correlated.

We next consider the case with $T = 4$, where we control the task similarity by changing Y_2 while fixing Y_1, Y_3 and Y_4. Here we let $(Y_1 \cup Y_2) \cap (Y_3 \cup Y_4) = \emptyset$ as in Section 4.2. As shown in Figure 2(b), forgetting surprisingly increases when task 1 and task 2 have more overlapping classes, which is also consistent with our analysis for the linear models. Indeed, this also justifies our observation that forgetting can decrease when the adjacent tasks are more dissimilar when studying the impact of task order.

5.2. Diversify the tasks in the early stage and order dissimilar tasks adjacently

We also evaluate the impact of task ordering on forgetting in CL with DNN, by constructing the tasks using a similar strategy as in Section 5.1. More specifically, we consider two different scenarios: (1) $T = 6$, where the task sequence includes one special task and five same tasks; (2) $T = 4$, where the task sequence includes two categories of tasks and each has two same tasks.

Figure 2(c) demonstrates forgetting in the first scenario w.r.t. the learning order of the special task, and three plots correspond to three different cases, respectively. It is clear that for all three cases, the optimal order of the special task to minimize forgetting is always in the first half of the sequence. For the second scenario, we evaluate forgetting in Figure 2(d) for all six possible task orders, where task indices 0 and 1 refer to the perfectly alternating order. We can see that the smallest forgetting is also achieved in the perfectly alternating order. These results indicate that our findings in Section 4.3 for the overparameterized linear models can also be carried over to CL with DNN, i.e., the optimal task order should diversify the tasks in the early stage and learn more different tasks adjacently. Such an implication is indeed consistent with the empirical observations in recent studies (Li et al., 2022; Bell & Lawrence, 2022). Note that in both Figure 2(c) and Figure 2(d), we normalize forgetting w.r.t. the worst forgetting in each case.

5.3. Weight the fresher old tasks more in forward knowledge transfer

Recently, there has been increasing interest in CL on leveraging task correlation to facilitate knowledge transfer (Ke et al., 2020; Lin et al., 2022b;a), which first selects the most correlated old tasks with the current task and then designs algorithms to directly increase the knowledge transfer between correlated tasks. By investigating knowledge transfer in the linear models, we show that improved algorithms can be motivated to achieve better knowledge transfer.

Given a task t in CL, the forward knowledge transfer (Veniat et al., 2020) in the linear model can be defined as

$$\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2] - \mathbb{E}[\|\boldsymbol{w}_t^r - \boldsymbol{w}_t^*\|^2], \tag{13}$$

where $\boldsymbol{w}_t^r$ is the learnt model of task t by starting from a random model. Intuitively, Equation (13) characterizes the gap in the testing performance between $\boldsymbol{w}_t$ learnt in CL and $\boldsymbol{w}_t^r$ learnt from scratch, for which a positive value means that the accumulated knowledge in CL benefits the learning of the current task. As the second term in Equation (13) is independent with CL, it suffices to analyze $\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2]$ for understanding the forward knowledge transfer. Based on Lemma B.2 (Appendix B), we can obtain

$$\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2] = r^t\|\boldsymbol{w}_t^*\|^2 + \sum_{i=1}^{t} \frac{nr^{t-i}}{p}\|\boldsymbol{w}_i^* - \boldsymbol{w}_t^*\|^2 + \frac{p\sigma^2}{p-n-1}.$$

While it is intuitive that better forward knowledge transfer can be achieved when $\|\boldsymbol{w}_i^* - \boldsymbol{w}_t^*\|^2$ is smaller for the current task t and the old task i, the impact of different old tasks on the current task is non-uniform, in the sense that a more recent old task i (i.e., $t - i$ is smaller) has a larger effect on the forward knowledge transfer to task t. This result implies that fresher old tasks should contribute more when designing algorithms to leverage correlated old tasks to facilitate better forward knowledge transfer.

To verify this insight, we consider the TRGP algorithm proposed in (Lin et al., 2022b). Specifically, TRGP first selects the most correlated old tasks with the current task and reuses their knowledge through a scaled weight projection to facilitate forward knowledge transfer, where all the selected old tasks are treated equivalently. We slightly modify TRGP by assigning a larger weight to the selected old task that is more recent to the current task, named as TRGP+, and evaluate its performance on standard CL benchmarks (PMNIST (Lopez-Paz & Ranzato, 2017) and Split CIFAR-100 (Krizhevsky et al., 2009)) and DNN architectures. As shown in Table 1, TRGP+ outperforms TRGP in both accuracy and forgetting. Assigning a larger weight to the more recent correlated old task not only improves the forward knowledge transfer, but also increases the backward knowledge transfer by forcing the learnt model of the current task to be closer to the model of those highly correlated old tasks.

Table 1. The averaged final testing accuracy (ACC) and backward transfer (BWT: negative value of forgetting, larger is better) over all the tasks on different datasets.

Method	PMNIST		Split CIFAR-100	
	ACC(%)	BWT(%)	ACC(%)	BWT(%)
TRGP	96.34	-0.8	74.46	-0.9
TRGP+	**96.75**	**-0.46**	**75.31**	**0.13**

6. Conclusions

In this work, we studied CL in the overparameterized linear models where each task is a linear regression problem and solved by using SGD. Under the assumption that each task has a sparse linear model with i.i.d. Gaussian features and noise, we derived the exact forms of both forgetting and generalization error, which built the key foundations of understanding the performance of CL. In particular, we investigated the impact of overparameterization, task similarity, and task ordering on both forgetting and generalization error. Experimental results on real datasets with DNNs indicated that our findings in linear models can even be carried over to CL in practice and leveraged to develop better algorithms.

Acknowledgements

The work has been partly supported by the U.S. National Science Foundation under the grants of NSF AI Institute (AI-EDGE) CNS-2112471, CNS-2106933, 2007231, CNS-1955535, CNS-1901057, ECCS-2113860, and CCF-1900145, and in part by Army Research Office under Grant W911NF-21-1-0244.

References

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Andle, J. and Yasaei Sekeh, S. Theoretical understanding of the information flow on continual learning performance. In *European Conference on Computer Vision*, pp. 86–101. Springer, 2022.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332, 2019.

Asanuma, H., Takagi, S., Nagano, Y., Yoshida, Y., Igarashi, Y., and Okada, M. Statistical mechanical analysis of catastrophic forgetting in continual learning with teacher and student networks. *Journal of the Physical Society of Japan*, 90(10):104001, 2021.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020a.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020b.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pp. 541–549. PMLR, 2018.

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019.

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

Bell, S. J. and Lawrence, N. D. The effect of task ordering in continual learning. *arXiv preprint arXiv:2205.13323*, 2022.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Bennani, M. A., Doan, T., and Sugiyama, M. Generalisation guarantees for continual learning with orthogonal gradient descent. *arXiv preprint arXiv:2006.11942*, 2020.

Cao, X., Liu, W., and Vempala, S. Provable lifelong learning of representations. In *International Conference on Artificial Intelligence and Statistics*, pp. 6334–6356. PMLR, 2022.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Chen, X., Papadimitriou, C., and Peng, B. Memory bounds for continual learning. *arXiv preprint arXiv:2204.10830*, 2022.

Doan, T., Bennani, M. A., Mazoure, B., Rabusseau, G., and Alquier, P. A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In *International Conference on Artificial Intelligence and Statistics*, pp. 1072–1080. PMLR, 2021.

Evron, I., Moroshko, E., Ward, R., Srebro, N., and Soudry, D. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, pp. 4028–4079. PMLR, 2022.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Goldfarb, D. and Hand, P. Analysis of catastrophic forgetting for random orthogonal transformation tasks in the overparameterized regime. In *International Conference*

on Artificial Intelligence and Statistics, pp. 2975–2993. PMLR, 2023.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

Jin, X., Sadhu, A., Du, J., and Ren, X. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34:29193–29205, 2021.

Ju, P., Lin, X., and Liu, J. Overfitting can be harmless for basis pursuit, but only to a degree. *Advances in Neural Information Processing Systems*, 33:7956–7967, 2020.

Ju, P., Lin, X., and Shroff, N. B. On the generalization power of overfitted two-layer neural tangent kernel models. *arXiv preprint arXiv:2103.05243*, 2021.

Ju, P., Lin, X., and Shroff, N. B. On the generalization power of the overfitted three-layer neural tangent kernel model. *arXiv preprint arXiv:2206.02047*, 2022.

Ke, Z., Liu, B., and Huang, X. Continual learning of a mixed sequence of similar and dissimilar tasks. *Advances in Neural Information Processing Systems*, 33:18493–18504, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

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

Li, Y., Li, M., Asif, M. S., and Oymak, S. Provable and efficient continual representation learning. *arXiv preprint arXiv:2203.02026*, 2022.

Lin, S., Yang, L., Fan, D., and Zhang, J. Beyond not-forgetting: Continual learning with backward knowledge transfer. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022a.

Lin, S., Yang, L., Fan, D., and Zhang, J. Trgp: Trust region gradient projection for continual learning. *Tenth International Conference on Learning Representations, ICLR 2022*, 2022b.

Liu, H. and Liu, H. Continual learning with recursive gradient optimization. *arXiv preprint arXiv:2201.12522*, 2022.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 2017.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Mitra, P. P. Understanding overfitting peaks in generalization error: Analytical risk curves for l_2 and l_1 penalized interpolation. *arXiv preprint arXiv:1906.03667*, 2019.

Muthukumar, V., Vodrahalli, K., and Sahai, A. Harmless interpolation of noisy data in regression. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2299–2303. IEEE, 2019.

Muthukumar, V., Vodrahalli, K., Subramanian, V., and Sahai, A. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1 (1):67–83, 2020.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Ramasesh, V. V., Dyer, E., and Raghu, M. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Saha, G., Garg, I., and Roy, K. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Satpathi, S. and Srikant, R. The dynamics of gradient descent for overparametrized neural networks. In *Learning for Dynamics and Control*, pp. 373–384. PMLR, 2021.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4548–4557. PMLR, 2018.

Veniat, T., Denoyer, L., and Ranzato, M. Efficient continual learning with modular networks and task-driven priors. *arXiv preprint arXiv:2012.12631*, 2020.

Yang, L., Lin, S., Zhang, J., and Fan, D. Grown: Grow only when necessary for continual learning. *arXiv preprint arXiv:2110.00908*, 2021.

Yin, D., Farajtabar, M., Li, A., Levine, N., and Mott, A. Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. *arXiv preprint arXiv:2006.10974*, 2020.

Yoon, J., Kim, S., Yang, E., and Hwang, S. J. Scalable and order-robust continual learning with additive parameter decomposition. In *Eighth International Conference on Learning Representations, ICLR 2020*. ICLR, 2020.

Supplemental Material

A. Experimental Details

A.1. Experimental details for Section 5.1 and Section 5.2

Datasets. We consider the MNIST dataset. For each task, we randomly select 200 samples for training and 1000 samples for testing. Different tasks have different subsets of classes.

DNN architecture and training details. We use a five-layer neural network with two convolutional layers and three fully-connected layers. Relu is used for the first four layers and Sigmoid is used for the last layer. The first convolutional layer is followed by 2D max-pooling operation with stride of 2. We learn each task by using SGD with a learning rate of 0.1 for 600 epochs. The forgetting and overall generalization error are evaluated as in Equation (6) and Equation (7), respectively, while here $\mathcal{L}_t(\boldsymbol{w})$ is defined as the mean-squared test error instead of Equation (5).

Task setups. For Figure 2(a), we consider the following setup:

- task 1: $(0, 1, 2, 3, 4)$.
- task 2: $(5, 6, 7, 8, 9)$, $(4, 5, 6, 7, 8)$, $(3, 4, 5, 6, 7)$, $(2, 3, 4, 5, 6)$, $(1, 2, 3, 4, 5)$, $(0, 1, 2, 3, 4)$, which correspond to the different numbers of overlapping classes with task 1.

For Figure 2(b), we randomly select three different setups:

- 'forgetting_0':
  - task 1: $(0, 1, 2)$.
  - task 2: $(3, 4, 5)$, $(2, 3, 4)$, $(1, 2, 3)$, $(0, 1, 2)$, which correspond to the different numbers of overlapping classes with task 1.
  - task 3: $(7, 8, 9)$.
  - task 4: $(7, 8, 9)$.
- 'forgetting_1':
  - task 1: $(3, 4, 5)$.
  - task 2: $(0, 1, 2)$, $(1, 2, 3)$, $(2, 3, 4)$, $(3, 4, 5)$, which correspond to the different numbers of overlapping classes with task 1.
  - task 3: $(6, 7, 8)$.
  - task 4: $(7, 8, 9)$.
- 'forgetting_2':
  - task 1: $(0, 1, 2)$.
  - task 2: $(7, 8, 9)$, $(2, 7, 8)$, $(1, 2, 7)$, $(0, 1, 2)$, which correspond to the different numbers of overlapping classes with task 1.
  - task 3: $(4, 5, 6)$.
  - task 4: $(4, 5, 6)$.

For Figure 2(c), we randomly select three different setups:

- 'forgetting_0': the special task is $(4, 5, 6, 7)$ and the other tasks are $(0, 1, 2, 3)$.
- 'forgetting_1': the special task is $(0, 1, 2, 3)$ and the other tasks are $(5, 6, 7, 8)$.
- 'forgetting_2': the special task is $(3, 4, 5, 6)$ and the other tasks are $(1, 2, 7, 8)$.

For Figure 2(d), we randomly select three different setups:

- 'forgetting_0': the two task categories are $(4, 5, 6, 7)$ and $(1, 2, 4, 5)$, and the task order indices are:
  - '0': $(4, 5, 6, 7)$, $(1, 2, 4, 5)$, $(4, 5, 6, 7)$, $(1, 2, 4, 5)$.
  - '1': $(1, 2, 4, 5)$, $(4, 5, 6, 7)$, $(1, 2, 4, 5)$, $(4, 5, 6, 7)$.
  - '2': $(4, 5, 6, 7)$, $(4, 5, 6, 7)$, $(1, 2, 4, 5)$, $(1, 2, 4, 5)$.
  - '3': $(1, 2, 4, 5)$, $(1, 2, 4, 5)$, $(4, 5, 6, 7)$, $(4, 5, 6, 7)$.
  - '4': $(4, 5, 6, 7)$, $(1, 2, 4, 5)$, $(1, 2, 4, 5)$, $(4, 5, 6, 7)$.
  - '5': $(1, 2, 4, 5)$, $(4, 5, 6, 7)$, $(4, 5, 6, 7)$, $(1, 2, 4, 5)$.
- 'forgetting_1': the two task categories are $(4, 5, 6, 7)$ and $(2, 3, 4, 5)$, and the task order indices are:
  - '0': $(4, 5, 6, 7)$, $(2, 3, 4, 5)$, $(4, 5, 6, 7)$, $(2, 3, 4, 5)$.
  - '1': $(2, 3, 4, 5)$, $(4, 5, 6, 7)$, $(2, 3, 4, 5)$, $(4, 5, 6, 7)$.
  - '2': $(4, 5, 6, 7)$, $(4, 5, 6, 7)$, $(2, 3, 4, 5)$, $(2, 3, 4, 5)$.
  - '3': $(2, 3, 4, 5)$, $(2, 3, 4, 5)$, $(4, 5, 6, 7)$, $(4, 5, 6, 7)$.
  - '4': $(4, 5, 6, 7)$, $(2, 3, 4, 5)$, $(2, 3, 4, 5)$, $(4, 5, 6, 7)$.
  - '5': $(2, 3, 4, 5)$, $(4, 5, 6, 7)$, $(4, 5, 6, 7)$, $(2, 3, 4, 5)$.
- 'forgetting_2': the two task categories are $(6, 7, 8, 9)$ and $(3, 4, 5, 6)$, and the task order indices are:
  - '0': $(6, 7, 8, 9)$, $(3, 4, 5, 6)$, $(6, 7, 8, 9)$, $(3, 4, 5, 6)$.
  - '1': $(3, 4, 5, 6)$, $(6, 7, 8, 9)$, $(3, 4, 5, 6)$, $(6, 7, 8, 9)$.
  - '2': $(6, 7, 8, 9)$, $(6, 7, 8, 9)$, $(3, 4, 5, 6)$, $(3, 4, 5, 6)$.
  - '3': $(3, 4, 5, 6)$, $(3, 4, 5, 6)$, $(6, 7, 8, 9)$, $(6, 7, 8, 9)$.
  - '4': $(6, 7, 8, 9)$, $(3, 4, 5, 6)$, $(3, 4, 5, 6)$, $(6, 7, 8, 9)$.
  - '5': $(3, 4, 5, 6)$, $(6, 7, 8, 9)$, $(6, 7, 8, 9)$, $(3, 4, 5, 6)$.

A.2. Experimental details for Section 5.3

A.2.1. TRGP vs TRGP+

TRGP (Lin et al., 2022b) seeks to solve the following optimization problem for the current task t:

$$\begin{aligned} \min_{\{\boldsymbol{w}^l\}_l, \{\boldsymbol{Q}^l_{j,t}\}_{l, j\in\mathcal{TR}^l_t}} \quad & \mathcal{L}(\{\boldsymbol{w}^l_{eff}\}_l, \mathcal{D}_t), \\ s.t \quad & \boldsymbol{w}^l_{eff} = \boldsymbol{w}^l + \sum\nolimits_{j\in\mathcal{TR}^l_t} [\mathrm{Proj}^Q_{S^l_j}(\boldsymbol{w}^l) - \mathrm{Proj}_{S^l_j}(\boldsymbol{w}^l)], \end{aligned} \tag{14}$$

where $\boldsymbol{w}^l$ is the DNN weight for the layer l, and S^l_j denotes the input subspace of the layer l for the old task $j < t$, which can be constructed by using SVD on the representation matrix for that layer. Two important designs are introduced in Equation (14):

- The trust region $\mathcal{TR}^l_t$: $\mathcal{TR}^l_t$ denotes the set of the most correlated old tasks selected for task t based on some correlation evaluation metric in a layer-wise manner. The purpose here is to select the most correlated old tasks and facilitate the forward knowledge transfer by reusing the learnt knowledge of the old tasks in $\mathcal{TR}^l_t$.
- The scaled weight projection $\mathrm{Proj}^Q_{S^l_j}(\boldsymbol{w}^l)$: $\mathrm{Proj}^Q_{S^l_j}(\boldsymbol{w}^l)$ is developed to reuse the learnt model of the selected old tasks in $\mathcal{TR}^l_t$. Specifically, for any $j \in \mathcal{TR}^l_t$,
$$\mathrm{Proj}^Q_{S^l_j}(\boldsymbol{w}^l) = \boldsymbol{w}^l_{t-1}\boldsymbol{B}^l_j\boldsymbol{Q}^l_{j,t}(\boldsymbol{B}^l_j)'$$
where $\boldsymbol{B}^l_j$ is the bases matrix for the subspace S^l_j, and $\boldsymbol{Q}^l_{j,t}$ is the scaling matrix to scale the weight projection onto S^l_j. In contrast, $\mathrm{Proj}_{S^l_j}(\boldsymbol{w}^l) = \boldsymbol{w}^l_{t-1}\boldsymbol{B}^l_j(\boldsymbol{B}^l_j)'$ is the standard weight projection onto S^l_j. Since the learnt knowledge for the old task j is indeed $\mathrm{Proj}_{S^l_j}(\boldsymbol{w}^l)$, scaling the projection provides a way to reuse this knowledge directly for learning the task t. Intuitively, $\mathrm{Proj}^Q_{S^l_j}(\boldsymbol{w}^l) - \mathrm{Proj}_{S^l_j}(\boldsymbol{w}^l)$ characterizes the boosted forward knowledge transfer from old task $j \in \mathcal{TR}^l_t$ to task t.

However, as shown in Equation (14), all the selected old tasks in $\mathcal{TR}_t^l$ are treated equivalently in the effective weight $\boldsymbol{w}_{eff}^l$, which could be suboptimal. As suggested by our theoretical results, we proposed a slightly modified version of TRGP, i.e., TRGP+, by assigning non-uniform weights for the most correlated old tasks selected in $\mathcal{TR}_t^l$:

$$\begin{aligned} \min_{\{\boldsymbol{w}^l\}_l, \{\boldsymbol{Q}_{j,t}^l\}_{l, j\in\mathcal{TR}_t^l}} \quad & \mathcal{L}(\{\boldsymbol{w}_{eff}^l\}_l, \mathcal{D}_t), \\ s.t \quad & \boldsymbol{w}_{eff}^l = \boldsymbol{w}^l + \sum\nolimits_{j\in\mathcal{TR}_t^l} \lambda_j [\mathrm{Proj}_{S_j^l}^Q(\boldsymbol{w}^l) - \mathrm{Proj}_{S_j^l}(\boldsymbol{w}^l)], \end{aligned} \tag{15}$$

where $\lambda_j > \lambda_{j'}$ if $t - j < t - j'$ for both $j, j' \in \mathcal{TR}_t^l$.

A.2.2. Experimental setup

Datasets. We consider two standard benchmarks in CL: (1) **PMNIST**: 10 sequential tasks will be created using different permutations, where each task has 10 classes; (2) **Split CIFAR-100**: The entire dataset of CIFAR-100 will be splitted into 10 group, where each task is a 10-way multi-class classification problem for each group.

DNN architecture and training details. Following (Lin et al., 2022b), we use a 3-layer fully-connected network with 2 hidden layer of 100 units for PMNIST, and train the network for 5 epochs with a batch size of 10 for each task. For Split CIFAR-100, we use a version of 5-layer AlexNet, and train the network for a maximum of 200 epochs with early stopping for each task. Two most correlated old tasks are selected for the current task for each layer, and we assign a larger weight of 1.2 to the more recent old task and 0.8 to the other one.

Evaluation metrics. The performance is evaluated based on ACC, the average final accuracy over all tasks, and Backward Transfer (BWT) which measures the forgetting of old tasks when learning new tasks. Specfically, ACC and BWT are defined as:

$$ACC = \frac{1}{T}\sum\nolimits_{i=1}^{T} A_{T,i}, BWT = \frac{1}{T-1}\sum\nolimits_{i=1}^{T-1} A_{T,i} - A_{i,i} \tag{16}$$

where $A_{T,i}$ is the accuracy of the model on i-th task after learning the T-th task sequentially.

B. Useful Lemmas

The following lemma characterizes the solution to the optimization problem Equation (4) for task t:

Lemma B.1. *The solution to the optimization problem Equation* (4)*, i.e., the learnt model for task t, is given by*

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \boldsymbol{X}_t(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}\left(\boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1}\right). \tag{17}$$

In the overparameterized case, multiple $\boldsymbol{w}_t$ exist to perfectly fit $(\boldsymbol{X}_t)^\top \boldsymbol{w} = \boldsymbol{y}_t$, and solving Equation (4) picks the one that has minimum l^2 distance to $\boldsymbol{w}_{t-1}$. Therefore, the solution in Equation (17) not only incorporates the information of current task t through $\mathcal{D}_t$ but also depends on the previous model evolution trajectory in CL.

By leveraging the recent advance in (Belkin et al., 2020), we can have the following lemma about the evolution of $\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2]$:

Lemma B.2. *Suppose $p \geq n+2$. For any task $t \in [1, T-1]$ and any old task $i \in [1, t]$, the following equation holds:*

$$\begin{aligned} & \mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2] \\ = & \left(1 - \frac{n}{p}\right)\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] + \frac{n}{p}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 + \frac{n\sigma^2}{p-n-1}. \end{aligned}$$

C. Additional Results

C.1. Characterization of negative forgetting

As shown in Figure 2(a), the forgetting can even be negative when the two tasks are positively correlated. Intuitively, because the common features play a similar role in these two tasks, task 2 updates the model in a favorable direction for task 1, which could even result in better performance of task 1 due to the backward knowledge transfer herein. A formal quantification of the condition for better performance of task 1 can be found in the following proposition:

Proposition C.1. *Suppose $\sigma^2 < \frac{p-n-1}{p}\|\boldsymbol{w}_1^*\|^2$ and $p \geq n+2$. The learning of task 2 would lead to a better model for task 1, i.e., $\mathbb{E}[F_2] \leq 0$, if*

$$2\langle \boldsymbol{w}_{1,\mathcal{S}_1}^*, \boldsymbol{w}_{2,\mathcal{S}_2}^*\rangle \geq \frac{n}{p}\|\boldsymbol{w}_1^*\|^2 + \|\boldsymbol{w}_2^*\|^2 + \frac{(p-n)\sigma^2}{p-n-1}.$$

C.2. Evolution of forgetting

We can also characterize the evolution of forgetting after learning new tasks. Based on the definition of forgetting, we have

$$\mathbb{E}[F_t] = \frac{1}{t-1}\sum_{i=1}^{t-1}\mathbb{E}\left[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2 - \|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2\right],$$

$$\mathbb{E}[F_{t+1}] = \frac{1}{t}\sum_{i=1}^{t}\mathbb{E}\left[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2 - \|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2\right].$$

Rearranging the above equations gives

$$\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] = (t-1)\mathbb{E}[F_t] + \sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2],$$

$$\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2] = t\mathbb{E}[F_{t+1}] + \sum_{i=1}^{t}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2] - \mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t^*\|^2].$$

Based on the relationship between $\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2]$ and $\mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2]$ characterized in Lemma B.2, it can be seen that

$$\begin{aligned}
&\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2] \\
=&t\mathbb{E}[F_{t+1}] + \sum_{i=1}^{t}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2] - \mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t^*\|^2] \\
=&\sum_{i=1}^{t-1}\left\{\left(1-\frac{n}{p}\right)\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] + \frac{n}{p}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 + \frac{n\sigma^2}{p-n-1}\right\} \\
=&\left(1-\frac{n}{p}\right)\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] + \frac{n}{p}\sum_{i=1}^{t-1}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 + \frac{n\sigma^2(t-1)}{p-n-1} \\
=&\left(1-\frac{n}{p}\right)\left\{(t-1)\mathbb{E}[F_t] + \sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2]\right\} + \frac{n}{p}\sum_{i=1}^{t-1}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 + \frac{n\sigma^2(t-1)}{p-n-1},
\end{aligned}$$

such that

$$\begin{aligned}
t\mathbb{E}[F_{t+1}] =&(t-1)\left(1-\frac{n}{p}\right)\mathbb{E}[F_t] + \left(1-\frac{n}{p}\right)\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2] + \frac{n}{p}\sum_{i=1}^{t-1}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 \\
&+ \frac{n\sigma^2(t-1)}{p-n-1} - \sum_{i=1}^{t}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2] + \mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t^*\|^2] \\
=&(t-1)\left(1-\frac{n}{p}\right)\mathbb{E}[F_t] - \frac{n}{p}\sum_{i=1}^{t-1}\mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2] - \mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2] \\
&+ \frac{n}{p}\sum_{i=1}^{t-1}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 + \mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t^*\|^2] + \frac{n\sigma^2(t-1)}{p-n-1}.
\end{aligned} \tag{18}$$

Let $i = t$ in Lemma B.2. We can show that

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t^*\|^2 - \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2] \\
=&\frac{n}{p}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_t^*\|^2 - \frac{n}{p}\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2] + \frac{n\sigma^2}{p-n-1}.
\end{aligned} \tag{19}$$

By substituting Equation (19) back to Equation (18), we can have

$$\mathbb{E}[F_{t+1}] = \frac{t-1}{t}\left(1-\frac{n}{p}\right)\mathbb{E}[F_t] + \frac{n}{tp}\sum_{i=1}^{t-1}\left\{\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2 - \mathbb{E}[\|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2]\right\}$$

$$+\frac{n}{tp}\left\{\|\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_t^*\|^2-\mathbb{E}[\|\boldsymbol{w}_t-\boldsymbol{w}_t^*\|^2]\right\}+\frac{n\sigma^2}{p-n-1}$$
$$=\frac{t-1}{t}\left(1-\frac{n}{p}\right)\mathbb{E}[F_t]+\frac{n}{tp}\sum_{i=1}^{t}\left\{\|\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_i^*\|^2-\mathbb{E}[\|\boldsymbol{w}_i-\boldsymbol{w}_i^*\|^2]\right\}+\frac{n\sigma^2}{p-n-1}. \tag{20}$$

C.3. Impact of overparameterization

1) Forgetting approaches zero with more parameters. In Equation (9), when $p\to\infty$, we have $r\to 1$, which implies that $(r^T-r^i)\to 0$ and $c_{i,j}\to 0$. Therefore, we can conclude that $\mathbb{E}[F_T]\to 0$ when $p\to\infty$. An intuitive explanation is that with more parameters, the model has a larger "memory" such that it can remember all knowledge of previous tasks, i.e., zero forgetting.

2) More parameters can alleviate the negative impact of task dissimilarity on generalization error. Term G2 in Equation (10) describes the effect of task dissimilarity on G_T. When $p\to\infty$, Term G2 approaches zero, which indicates that the negative impact of task dissimilarity on generalization error diminishes. In some special cases, we can further show that Term G2 is monotonically decreasing with respect to p, e.g., $T=2$ shown in Equation (12). A more general[4] case is when $\sum_{k=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2=C$ for all task i, we have Term G2 $=\frac{1-r^T}{T}C$ which is also monotonically decreasing w.r.t. p.

C.4. Impact of task order

(1) **[Special case III]** There are three categories (C_1, C_2 and C_3) of tasks: each category contains the same number of tasks; the tasks are same in the same category but different across categories. Without loss of generality, we assume that for any task i and j

$$\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2=\begin{cases}0, & \text{if } i,j\in C_m \text{ for } m\in\{1,2,3\};\\ 1, & \text{else.}\end{cases}$$

Based on Theorem 4.1, we can show that the optimal task order for Special case III follows a similar structure of that for Special case II, as characterized in the following proposition:

Proposition C.2. *Suppose $p\geq n+2$. For $T=6$, the optimal task order to minimize forgetting is the perfectly alternating order, i.e., $(C_i,C_j,C_k,C_i,C_j,C_k)$, where $i,j,k\in\{1,2,3\}$, $i\neq j$, $i\neq k$ and $j\neq k$.*

(2) **[The optimal task order can be different for minimizing forgetting and generalization error]**

[Special case I] As shown in Proposition 4.4, the optimal task order to minimize forgetting is to learn the special task between the $2nd$ place and the $\frac{T}{2}th$ place. In stark contrast, this special task, which has the largest value of $\sum_{k=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2$, should be always learnt in the very first place in order to minimize the generalization error, i.e., $i=1$. The underlying rationale is that the generalization error characterizes the average testing performance of the final model on all tasks, which is maximized when the final model works the best for the majority. Therefore, in this case the optimal order for minimizing forgetting is different from that for minimizing generalization error.

[Special case II] As shown in Proposition 4.5, the optimal task order to minimize forgetting is the perfectly alternating order. In contrast, the task order indeed does not affect the generalization performance, because $\sum_{k=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2$ is same for every task $i\in\mathbb{T}$. In this case, the optimal task order for minimizing forgetting is also 'optimal' for minimizing generalization error. That is to say, we can find an optimal task order to minimize forgetting and generalization error simultaneously.

D. Proofs

D.1. Proof of Lemma B.1

Let $\hat{\boldsymbol{w}}=\boldsymbol{w}-\boldsymbol{w}_{t-1}$. It is clear that Equation (4) can be reformulated as

$$\begin{aligned}&\min\ \|\hat{\boldsymbol{w}}\|_2,\\ &s.t.\ \ \boldsymbol{X}_t^\top\hat{\boldsymbol{w}}=\boldsymbol{y}_t-\boldsymbol{X}_t^\top\boldsymbol{w}_{t-1}.\end{aligned} \tag{21}$$

[4]For general T, this requirement holds if the ground truth of each task has the same power and is orthogonal to each other, i.e., $\|\boldsymbol{w}_i^*\|^2=\|\boldsymbol{w}_j^*\|^2$ and $(\boldsymbol{w}_i^*)^T\boldsymbol{w}_j^*=0$ for all $i\neq j$.

For the overparameterized case, $\boldsymbol{X}_t^\top \boldsymbol{X}_t$ is invertible. Using the Lagrange multipliers, we can get

$$\min_{\hat{\boldsymbol{w}},\lambda} \frac{\hat{\boldsymbol{w}}^\top \hat{\boldsymbol{w}}}{2} + \lambda^T[\boldsymbol{X}_t^\top \hat{\boldsymbol{w}} - (\boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1})].$$

By setting the derivative w.r.t. $\hat{\boldsymbol{w}}$ to 0, it follows that

$$\hat{\boldsymbol{w}}^* = -\boldsymbol{X}_t \lambda \tag{22}$$

such that

$$\boldsymbol{X}_t^\top \hat{\boldsymbol{w}}^* = -\boldsymbol{X}_t^\top \boldsymbol{X}_t \lambda = \boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1}.$$

Therefore,

$$\lambda = -(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}(\boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1}). \tag{23}$$

By substituting Equation (23) into Equation (22), we can have

$$\hat{\boldsymbol{w}}^* = \boldsymbol{X}_t(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}(\boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1})$$

such that

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \boldsymbol{X}_t(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}(\boldsymbol{y}_t - \boldsymbol{X}_t^\top \boldsymbol{w}_{t-1}).$$

D.2. Proof of Lemma B.2

Let $\boldsymbol{P}_t \coloneqq \boldsymbol{X}_t(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}\boldsymbol{X}_t^\top$ and $\boldsymbol{X}_t^\dagger \coloneqq \boldsymbol{X}_t(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}$ for any $t \in \mathbb{T}$, where $\boldsymbol{P}_t$ characterizes the projection onto the row space of $\boldsymbol{X}_t^\top$. Based on Lemma B.1, we have

$$\boldsymbol{w}_{t+1} = (\boldsymbol{I} - \boldsymbol{P}_{t+1})\boldsymbol{w}_t + \boldsymbol{P}_{t+1}\boldsymbol{w}_{t+1}^* + \boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1}. \tag{24}$$

Intuitively, the learnt model $\boldsymbol{w}_{t+1}$ for task $t+1$ is an 'interpolation' between the learnt model $\boldsymbol{w}_t$ for task t and the optimal task model $\boldsymbol{w}_{t+1}^*$ for task $t+1$, while being perturbed by the random noise z_{t+1}.

Let $H = (\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*) + \boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)$. Based on Equation (24), we can know that

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_i^*\|^2] \\
=&\mathbb{E}[\|(\boldsymbol{I} - \boldsymbol{P}_{t+1})\boldsymbol{w}_t + \boldsymbol{P}_{t+1}\boldsymbol{w}_{t+1}^* + \boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1} - \boldsymbol{w}_i^*\|^2] \\
=&\mathbb{E}[\|(\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*) + \boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*) + \boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1}\|^2] \\
=&\mathbb{E}[\|H + \boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1}\|^2] \\
=&\underbrace{\mathbb{E}[\|H\|^2]}_{(a)} + \underbrace{2\mathbb{E}[\langle H, \boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1}\rangle]}_{(b)} + \underbrace{\mathbb{E}[\|\boldsymbol{X}_{t+1}^\dagger \boldsymbol{z}_{t+1}\|^2]}_{(c)}.
\end{aligned} \tag{25}$$

(1) For the term (a), we have

$$\begin{aligned}
\mathbb{E}[\|H\|^2] =&\mathbb{E}[\|(\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*) + \boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\|^2] \\
=&\mathbb{E}[\|(\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*)\|^2] + \mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\|^2] + 2\mathbb{E}[\langle(\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*), \boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\rangle] \\
\overset{(a)}{=}&\mathbb{E}[\|(\boldsymbol{I} - \boldsymbol{P}_{t+1})(\boldsymbol{w}_t - \boldsymbol{w}_i^*)\|^2] + \mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\|^2] \\
\overset{(b)}{=}&\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] - \mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_t - \boldsymbol{w}_i^*)\|^2] + \mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\|^2]
\end{aligned} \tag{26}$$

where (a) is because of the orthogonality between $\boldsymbol{I} - \boldsymbol{P}_{t+1}$ and $\boldsymbol{P}_{t+1}$, and (b) is due to the Pythagorean theorem.

Because $\boldsymbol{P}_{t+1}$ is the orthogonal projection matrix for the row space of $\boldsymbol{X}_{t+1}$, based on the rotational symmetry of the standard normal distribution, it follows that

$$\mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*)\|^2] = \frac{n}{p}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2, \tag{27}$$

and

$$\mathbb{E}[\|\boldsymbol{P}_{t+1}(\boldsymbol{w}_t - \boldsymbol{w}_i^*)\|^2] = \frac{n}{p}\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2], \tag{28}$$

since $\boldsymbol{P}_{t+1}$ is independent with $\boldsymbol{w}_t$.

By substituting Equation (27) and Equation (28) back to Equation (26), we can obtain that

$$\mathbb{E}[\|H\|^2] = \left(1 - \frac{n}{p}\right)\mathbb{E}[\|\boldsymbol{w}_t - \boldsymbol{w}_i^*\|^2] + \frac{n}{p}\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}_i^*\|^2. \tag{29}$$

(2) For the term (b), we have

$$\begin{aligned}\mathbb{E}[\langle H, \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] =& \mathbb{E}[\langle(\boldsymbol{I}-\boldsymbol{P}_{t+1})(\boldsymbol{w}_t-\boldsymbol{w}_i^*)+\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_i^*), \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] \\ =& \mathbb{E}[\langle(\boldsymbol{I}-\boldsymbol{P}_{t+1})(\boldsymbol{w}_t-\boldsymbol{w}_i^*), \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] + \mathbb{E}[\langle\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_i^*), \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle].\end{aligned}$$

Because $(\boldsymbol{I}-\boldsymbol{P}_{t+1})$ is the projection onto the null space of $\boldsymbol{X}_{t+1}^{\top}$ and $\boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}$ is a vector in the row space of $\boldsymbol{X}_{t+1}^{\top}$, it follows that

$$\mathbb{E}[\langle(\boldsymbol{I}-\boldsymbol{P}_{t+1})(\boldsymbol{w}_t-\boldsymbol{w}_i^*), \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] = 0. \tag{30}$$

And since

$$\mathbb{E}[\langle\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_i^*), \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] = \mathbb{E}[\langle(\boldsymbol{X}_{t+1}^{\dagger})^{\top}\boldsymbol{P}_{t+1}(\boldsymbol{w}_{t+1}^*-\boldsymbol{w}_i^*), \boldsymbol{z}_{t+1}\rangle] = 0.$$

we can know that

$$\mathbb{E}[\langle H, \boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\rangle] = 0. \tag{31}$$

(3) For the term (c), we apply the "trace trick" by following (Belkin et al., 2020). Specifically, it can be first seen that

$$\begin{aligned}\|\boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\|^2 =& \|\boldsymbol{X}_{t+1}(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}\boldsymbol{z}_{t+1}\|^2 \\ =& tr((\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}\boldsymbol{z}_{t+1}\boldsymbol{z}_{t+1}^{\top}) \\ =& tr((\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}\boldsymbol{z}_{t+1}\boldsymbol{z}_{t+1}^{\top})\end{aligned}$$

Due to the independence between $\boldsymbol{X}_{t+1}$ and the random noise $\boldsymbol{z}_{t+1}$, we can have that

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\|^2] =& \mathbb{E}[tr((\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}\boldsymbol{z}_{t+1}\boldsymbol{z}_{t+1}^{\top}))] \\ =& tr[\mathbb{E}[(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}\boldsymbol{z}_{t+1}\boldsymbol{z}_{t+1}^{\top}]] \\ =& tr(\mathbb{E}[(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}]\mathbb{E}[\boldsymbol{z}_{t+1}\boldsymbol{z}_{t+1}^{\top}]) \\ =& \sigma^2 tr(\mathbb{E}[(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}]).\end{aligned}$$

Since $(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}$ follows the inverse-Wishart distribution with identity scale matrix $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ and p degrees-of-freedom, and each diagonal entry of $(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}$ has a reciprocal that follows the χ^2 distribution with $p-n+1$ degrees-of-freedom. Therefore, for $p \geq n+2$,

$$tr(\mathbb{E}[(\boldsymbol{X}_{t+1}^{\top}\boldsymbol{X}_{t+1})^{-1}]) = \frac{n}{p-n+1},$$

such that

$$\mathbb{E}[\|\boldsymbol{X}_{t+1}^{\dagger}\boldsymbol{z}_{t+1}\|^2] = \frac{n\sigma^2}{p-n+1}. \tag{32}$$

Lemma B.2 can be proved by substituting Equation (29), Equation (31) and Equation (32) to Equation (25).

D.3. Proof of Theorem 4.1

Based on Lemma B.2, we can have that

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{w}_t-\boldsymbol{w}_i^*\|^2] =& \left(1-\frac{n}{p}\right)^t\|\boldsymbol{w}_0-\boldsymbol{w}_i^*\|^2 + \sum_{k=1}^{t}\left(1-\frac{n}{p}\right)^{t-k}\frac{n}{p}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2 \\ &+ \frac{n\sigma^2}{p-n-1}\sum_{k=1}^{t}\left(1-\frac{n}{p}\right)^{t-k} \\ =& \left(1-\frac{n}{p}\right)^t\|\boldsymbol{w}_i^*\|^2 + \sum_{k=1}^{t}\left(1-\frac{n}{p}\right)^{t-k}\frac{n}{p}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2 \\ &+ \frac{n\sigma^2}{p-n-1}\sum_{k=1}^{t}\left(1-\frac{n}{p}\right)^{t-k} \quad (\text{since } \boldsymbol{w}_0=\boldsymbol{0}).\end{aligned} \tag{33}$$

Let $t=i$ in Equation (33). We have

$$\mathbb{E}[\|\boldsymbol{w}_i-\boldsymbol{w}_i^*\|^2] = \left(1-\frac{n}{p}\right)^i\|\boldsymbol{w}_i^*\|^2 + \sum_{k=1}^{i}\left(1-\frac{n}{p}\right)^{i-k}\frac{n}{p}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2$$

$$+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{i}\left(1-\frac{n}{p}\right)^{i-k}. \tag{34}$$

Based on Equation (33) and Equation (34), we can obtain the closed form of $\mathbb{E}[F_T]$:

$$\begin{aligned}
&\mathbb{E}[F_T]\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\mathbb{E}\left[\|\boldsymbol{w}_T-\boldsymbol{w}_i^*\|^2-\|\boldsymbol{w}_i-\boldsymbol{w}_i^*\|^2\right]\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left(1-\frac{n}{p}\right)^T\|\boldsymbol{w}_i^*\|^2+\sum_{k=1}^{T}\left(1-\frac{n}{p}\right)^{T-k}\frac{n}{p}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{T}\left(1-\frac{n}{p}\right)^{T-k}\\
&-\left(1-\frac{n}{p}\right)^i\|\boldsymbol{w}_i^*\|^2-\sum_{k=1}^{i}\left(1-\frac{n}{p}\right)^{i-k}\frac{n}{p}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2-\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{i}\left(1-\frac{n}{p}\right)^{i-k}\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{k=1}^{i}\frac{n}{p}\left[\left(1-\frac{n}{p}\right)^{T-k}-\left(1-\frac{n}{p}\right)^{i-k}\right]\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2\\
&+\sum_{k=i+1}^{T}\frac{n}{p}\left(1-\frac{n}{p}\right)^{T-k}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{i}\left[\left(1-\frac{n}{p}\right)^{T-k}-\left(1-\frac{n}{p}\right)^{i-k}\right]\\
&+\frac{n\sigma^2}{p-n-1}\sum_{k=i+1}^{T}\left(1-\frac{n}{p}\right)^{T-k}\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2\\
&+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{i}\left[\left(1-\frac{n}{p}\right)^{T-k}-\left(1-\frac{n}{p}\right)^{i-k}\right]+\frac{n\sigma^2}{p-n-1}\sum_{k=i+1}^{T}\left(1-\frac{n}{p}\right)^{T-k}\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2\\
&+\frac{n\sigma^2}{p-n-1}\left[\sum_{k=1}^{T}\left(1-\frac{n}{p}\right)^{T-k}-\sum_{k=1}^{i}\left(1-\frac{n}{p}\right)^{i-k}\right]\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2\\
&+\frac{n\sigma^2}{p-n-1}\left[\frac{1-\left(1-\frac{n}{p}\right)^T}{1-\left(1-\frac{n}{p}\right)}-\frac{1-\left(1-\frac{n}{p}\right)^i}{1-\left(1-\frac{n}{p}\right)}\right]\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2\\
&+\frac{n\sigma^2}{p-n-1}\frac{p}{n}\left[\left(1-\left(1-\frac{n}{p}\right)^T\right)-\left(1-\left(1-\frac{n}{p}\right)^i\right)\right]\Bigg\}\\
=&\frac{1}{T-1}\sum_{i=1}^{T-1}\Bigg\{\left[\left(1-\frac{n}{p}\right)^T-\left(1-\frac{n}{p}\right)^i\right]\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2\\
&+\frac{p\sigma^2}{p-n-1}\left[\left(1-\frac{n}{p}\right)^i-\left(1-\frac{n}{p}\right)^T\right]\Bigg\}
\end{aligned}$$

$$=\frac{1}{T-1}\sum_{i=1}^{T-1}\left\{(r^T-r^i)\|\boldsymbol{w}_i^*\|^2+\sum_{j>i}^{T}c_{i,j}\|\boldsymbol{w}_i^*-\boldsymbol{w}_j^*\|^2+\frac{p\sigma^2}{p-n-1}\left(r^i-r^T\right)\right\}.$$

Based on Equation (33), we can also obtain the exact form of the generalization error. Specifically,

$$\begin{aligned}&\mathbb{E}[\|\boldsymbol{w}_T-\boldsymbol{w}_i^*\|^2]\\=&\left(1-\frac{n}{p}\right)^T\|\boldsymbol{w}_i^*\|^2+\sum_{k=1}^{T}\frac{n}{p}\left(1-\frac{n}{p}\right)^{T-k}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{T}\left(1-\frac{n}{p}\right)^{T-k},\end{aligned}$$

such that

$$\begin{aligned}\mathbb{E}[G_T]=&\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}[\|\boldsymbol{w}_T-\boldsymbol{w}_i^*\|^2]\\=&\frac{1}{T}\left(1-\frac{n}{p}\right)^T\sum_{i=1}^{T}\|\boldsymbol{w}_i^*\|^2+\frac{1}{T}\sum_{k=1}^{T}\frac{n}{p}\left(1-\frac{n}{p}\right)^{T-k}\sum_{i=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2\\&+\frac{n\sigma^2}{p-n-1}\sum_{k=1}^{T}\left(1-\frac{n}{p}\right)^{T-k}\\=&\frac{1}{T}\left(1-\frac{n}{p}\right)^T\sum_{i=1}^{T}\|\boldsymbol{w}_i^*\|^2+\frac{1}{T}\sum_{k=1}^{T}\frac{n}{p}\left(1-\frac{n}{p}\right)^{T-k}\sum_{i=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2\\&+\frac{n\sigma^2}{p-n-1}\frac{1-\left(1-\frac{n}{p}\right)^T}{1-\left(1-\frac{n}{p}\right)}\\=&\frac{1}{T}\left(1-\frac{n}{p}\right)^T\sum_{i=1}^{T}\|\boldsymbol{w}_i^*\|^2+\frac{1}{T}\sum_{k=1}^{T}\frac{n}{p}\left(1-\frac{n}{p}\right)^{T-k}\sum_{i=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2\\&+\frac{p\sigma^2}{p-n-1}\left[1-\left(1-\frac{n}{p}\right)^T\right]\\=&\frac{r^T}{T}\sum_{i=1}^{T}\|\boldsymbol{w}_i^*\|^2+\frac{1}{T}\sum_{i=1}^{T}\frac{nr^{T-i}}{p}\sum_{k=1}^{T}\|\boldsymbol{w}_k^*-\boldsymbol{w}_i^*\|^2+\frac{p\sigma^2}{p-n-1}\left(1-r^T\right).\end{aligned}$$

D.4. Proof of Proposition C.1

Based on Theorem 4.1, it follows that

$$\begin{aligned}\mathbb{E}[F_2]=&(r^2-r)\|\boldsymbol{w}_1^*\|^2+\frac{n}{p}\|\boldsymbol{w}_1^*-\boldsymbol{w}_2^*\|^2+\frac{nr\sigma^2}{p-n-1}\\=&-\left(1-\frac{n}{p}\right)\frac{n}{p}\|\boldsymbol{w}_{1,s}^*\|^2+\frac{n}{p}\|\boldsymbol{w}_{1,s}^*\|^2+\frac{n}{p}\|\boldsymbol{w}_{2,s}^*\|^2-2\frac{n}{p}\langle\boldsymbol{w}_{1,s}^*,\boldsymbol{w}_{2,s}^*\rangle+\frac{nr\sigma^2}{p-n-1}\\=&\left(\frac{n}{p}\right)^2\|\boldsymbol{w}_{1,s}^*\|^2+\frac{n}{p}\|\boldsymbol{w}_{2,s}^*\|^2-2\frac{n}{p}\langle\boldsymbol{w}_{1,s}^*,\boldsymbol{w}_{2,s}^*\rangle+\frac{nr\sigma^2}{p-n-1}.\end{aligned}$$

When $\sigma^2<\frac{p-n-1}{p}\|\boldsymbol{w}_1^*\|^2$,

$$\frac{n}{p}\|\boldsymbol{w}_1^*\|^2+\|\boldsymbol{w}_2^*\|^2+\frac{(p-n)\sigma^2}{p-n-1}\leq\|\boldsymbol{w}_1^*\|^2+\|\boldsymbol{w}_2^*\|^2,$$

such that $\mathbb{E}[F_2]\leq 0$ if

$$2\langle\boldsymbol{w}_{1,\mathcal{S}_1}^*,\boldsymbol{w}_{2,\mathcal{S}_2}^*\rangle\geq\frac{n}{p}\|\boldsymbol{w}_1^*\|^2+\|\boldsymbol{w}_2^*\|^2+\frac{(p-n)\sigma^2}{p-n-1}.$$

D.5. Proof of Proposition 4.4

Without loss of generality, we assume that $\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\| = 1$ for task i in Category I and task j in Category II. It follows that

$$\begin{aligned}\tilde{F}_T(\boldsymbol{w}_T) =& \sum_{i<i^*} c_{i,i^*} + \sum_{j>i^*} c_{i^*,j} \\ =&(1-r)\left(\sum_{i=1}^{i^*-1}(r^{T-i} - r^{i^*-i} + r^{T-i^*}) + \sum_{j=i^*+1}^{T}(r^{T-i^*} - r^{j-i^*} + r^{T-j})\right) \\ =&(1-r)\left((T-1)\cdot r^{T-i^*} + r^{T-i^*+1}\frac{r^{i^*-1}-1}{r-1} - r\frac{r^{i^*-1}-1}{r-1} + 1 - r^{T-i^*}\right) \\ =&(1-r)(T-2)r^{T-i^*} + (r^{T-i^*}-1)(1-r^{i^*-1})r + (1-r).\end{aligned}$$

Letting $\alpha \coloneqq r^{T-i^*}$. Then minimizing $\tilde{F}_T(\boldsymbol{w}_T)$ is equivalent to minimize

$$\begin{aligned}&(1-r)(T-2)\alpha + (\alpha-1)(1-\frac{r^{T-1}}{\alpha})r \\ =&((1-r)(T-2)+r)\alpha + \frac{r^T}{\alpha} - r^T - r.\end{aligned}$$

By setting the derivative w.r.t. α to 0, we can have that the optimal value of α is

$$\alpha = \sqrt{\frac{r^T}{T-2-(T-1)r}} \tag{35}$$

which is clearly increasing with r. Therefore, the optimal order of the special task i^* is non-increasing with r, i.e., non-decreasing with $\frac{n}{p}$.

D.6. Proof of Proposition 4.5

Without loss of generality, we assume that for any task i and j

$$\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2 = \begin{cases} 0, & \text{if task } i \text{ and } j \text{ are in the same category;} \\ 1, & \text{if task } i \text{ and } j \text{ are in the different categories.}\end{cases}$$

Based on the closed form of forgetting, we can see that it suffices to minimize $\sum_{i=1}^{T-1}\sum_{j>i}^{T} c_{i,j}\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$ in order to minimize the forgetting $F_T(\boldsymbol{w}_T)$, where $c_{i,j} = (1-r)(r^{T-i} - r^{j-i} + r^{T-j})$. Besides, since whenever we change the order between the i-th task and the j-th task, the value of $r^{T-i} + r^{T-j}$ does not change. In other words, only the term r^{j-i} affects the optimal task order, which should minimize $\sum_{i=1}^{T-1}\sum_{j>i}^{T}(-r^{j-i})\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$.

(1) For the case $T = 4$, there are three effective task orders: (1) task $1 \in C_1$, task $2 \in C_1$, task $3 \in C_2$, task $4 \in C_2$ ((C_1, C_1, C_2, C_2) for simplicity); (2) (C_1, C_2, C_1, C_2); (3) (C_1, C_2, C_2, C_1). Swapping all tasks in C_1 with all tasks in C_2 does not change the value of forgetting, e.g., (C_1, C_1, C_2, C_2) has the same forgetting with (C_2, C_2, C_1, C_1). In what follows, we compare $\sum_{i=1}^{T-1}\sum_{j>i}^{T}(-r^{j-i})\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2$ among these three orders.

(a) For (C_1, C_1, C_2, C_2),

$$\sum_{i=1}^{T-1}\sum_{j>i}^{T}(-r^{j-i})\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2 = -(r^2 + r^3 + r + r^2).$$

(b) For (C_1, C_2, C_1, C_2),

$$\sum_{i=1}^{T-1}\sum_{j>i}^{T}(-r^{j-i})\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2 = -(r + r^3 + r + r).$$

(c) For (C_1, C_2, C_2, C_1),

$$\sum_{i=1}^{T-1}\sum_{j>i}^{T}(-r^{j-i})\|\boldsymbol{w}_i^* - \boldsymbol{w}_j^*\|^2 = -(r + r^2 + r + r^2).$$

It is clear that the alternating task order, i.e., (C_1, C_2, C_1, C_2) and (C_2, C_1, C_2, C_1), is the optimal order for this special

case.

(2) For the case $T = 6$, based on the closed form of forgetting in Theorem 4.1, we can use computer programming to show that besides the perfectly alternating task order, i.e., $(C_1, C_2, C_1, C_2, C_1, C_2)$ and $(C_2, C_1, C_2, C_1, C_2, C_1)$, there are 10 effective task orders as illustrated in Table 2. We further evaluate the difference of forgetting between each task order in Table 2 and the perfectly alternating task order, where a positive difference means that the corresponding task order will lead a larger forgetting than the perfectly alternating task order. It can be verified that the difference of forgetting is positive for all the task orders in Table 2, which indicates that the optimal task order is the perfectly alternating task order.

Index	Order	Difference of forgetting
1	$(C_1, C_2, C_1, C_2, C_1, C_2)$	0
2	$(C_1, C_1, C_2, C_1, C_2, C_2)$	$r\left(2 - 2r + 2r^2 - 2r^3\right)$
3	$(C_1, C_1, C_2, C_2, C_1, C_2)$	$r\left(2 - 3r + 2r^2 - r^3\right)$
4	$(C_1, C_1, C_2, C_2, C_2, C_1)$	$r\left(3 - 3r - r^3 + r^4\right)$
5	$(C_1, C_2, C_2, C_1, C_1, C_2)$	$r\left(2 - 4r + 2r^2\right)$
6	$(C_1, C_2, C_2, C_1, C_2, C_1)$	$r\left(1 - 2r + 2r^2 - 2r^3 + r^4\right)$
7	$(C_1, C_1, C_1, C_2, C_2, C_2)$	$r\left(4 - 2r - 2r^3\right)$
8	$(C_1, C_2, C_1, C_2, C_2, C_1)$	$r\left(1 - 2r + 2r^2 - 2r^3 + r^4\right)$
9	$(C_1, C_2, C_1, C_1, C_2, C_2)$	$r\left(2 - 3r + 2r^2 - r^3\right)$
10	$(C_1, C_2, C_2, C_2, C_1, C_1)$	$r\left(3 - 3r - r^3 + r^4\right)$

Table 2. Evaluation of the difference of forgetting between each effective task order and the perfectly alternating task order $(C_1, C_2, C_1, C_2, C_1, C_2)$, where a positive difference means that the corresponding task order will lead a larger forgetting than the perfectly alternating task order.

D.7. Proof of Proposition C.2

Following the same strategy with Special case II, we can have Table 3 to show all effective task orders and their difference of forgetting with the perfectly alternating task order, i.e., $(C_1, C_2, C_3, C_1, C_2, C_3)$ and its 'equivalent' task orders (e.g., $(C_1, C_3, C_2, C_1, C_3, C_2)$). It can also be verified that the perfectly alternating task order is the optimal task order in this case.

Index	Order	Difference of forgetting
1	$(C_1, C_2, C_3, C_1, C_2, C_3)$	0
2	$(C_1, C_2, C_1, C_2, C_3, C_3)$	$r\left(1 + 2r - 3r^2\right)$
3	$(C_1, C_2, C_2, C_3, C_3, C_1)$	$r\left(2 - 3r^2 + r^4\right)$
4	$(C_1, C_2, C_1, C_3, C_2, C_3)$	$r^2\left(2 - 2r\right)$
5	$(C_1, C_2, C_3, C_2, C_1, C_3)$	$r^2\left(1 - 2r + r^2\right)$
6	$(C_1, C_2, C_3, C_1, C_3, C_2)$	$r^2\left(1 - 2r + r^2\right)$
7	$(C_1, C_1, C_2, C_3, C_2, C_3)$	$r\left(1 + 2r - 3r^2\right)$
8	$(C_1, C_2, C_2, C_1, C_3, C_3)$	$r\left(2 - 2r^2\right)$
9	$(C_1, C_1, C_2, C_2, C_3, C_3)$	$r\left(3 - 3r^2\right)$
10	$(C_1, C_2, C_1, C_3, C_3, C_2)$	$r\left(1 + r - 3r^2 + r^3\right)$
11	$(C_1, C_2, C_3, C_3, C_1, C_2)$	$r\left(1 - 3r^2 + 2r^3\right)$
12	$(C_1, C_2, C_3, C_3, C_2, C_1)$	$r\left(1 - 2r^2 + r^4\right)$
13	$(C_1, C_2, C_2, C_3, C_1, C_3)$	$r\left(1 + r - 3r^2 + r^3\right)$
14	$(C_1, C_1, C_2, C_3, C_3, C_2)$	$r\left(2 - 2r^2\right)$
15	$(C_1, C_2, C_3, C_2, C_3, C_1)$	$r^2\left(2 - 3r + r^3\right)$

Table 3. Evaluation of the difference of forgetting between each effective task order and the perfectly alternating task order $(C_1, C_2, C_3, C_1, C_2, C_3)$, where a positive difference means that the corresponding task order will lead a larger forgetting than the perfectly alternating task order.

D.8. Proof of Theorem 4.3

Intuitive explanation of Theorem 4.3: In the underparameterized region, minimizing the loss Equation (3) for the current task t will lead to a unique solution for this task, which does not depend on the learning process and the learned model of previous tasks. That is to say, the task learning is independent among all tasks, such that (i) the learning order of the first $T-1$ tasks does not matter, and (ii) both forgetting and generalization performance depend only on the model distance between the last task and the other tasks, i.e., $\sum_{i=1}^{T-1} \|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2$.

Now we formally prove Theorem 4.3.

For the underparameterized regime, the solution of minimizing the training loss is

$$\begin{aligned}\boldsymbol{w}_t =&(\boldsymbol{X}_t\boldsymbol{X}_t^\top)^{-1}\boldsymbol{X}_t\boldsymbol{y}_t\\ =&(\boldsymbol{X}_t\boldsymbol{X}_t^\top)^{-1}\boldsymbol{X}_t\left(\boldsymbol{X}_t^\top\boldsymbol{w}_t^* + \boldsymbol{z}_t\right)\\ =&\boldsymbol{w}_t^* + (\boldsymbol{X}_t\boldsymbol{X}_t^\top)^{-1}\boldsymbol{X}_t\boldsymbol{z}_t.\end{aligned}$$

It follows that

$$\boldsymbol{w}_T - \boldsymbol{w}_i^* = \boldsymbol{w}_T^* - \boldsymbol{w}_i^* + (\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}\boldsymbol{X}_T\boldsymbol{z}_T,$$

such that the model error for the i-th task can be represented as:

$$\|\boldsymbol{w}_T - \boldsymbol{w}_i^*\|^2 = \|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2 + \|(\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}\boldsymbol{X}_T\boldsymbol{z}_T\|^2.$$

By taking expectation on both sides, we can have

$$\mathbb{E}\|\boldsymbol{w}_T - \boldsymbol{w}_i^*\|^2 = \|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2 + \frac{p\sigma^2}{n-p-1}.$$

Therefore, it can be shown that

$$\mathbb{E}[G_T] = \mathbb{E}\frac{1}{T}\sum_{i=1}^{T}\|\boldsymbol{w}_T - \boldsymbol{w}_i^*\|^2 = \left(\frac{1}{T}\sum_{i=1}^{T}\|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2\right) + \frac{p\sigma^2}{n-p-1}$$

and

$$\begin{aligned}\mathbb{E}[F_T] =&\frac{1}{T-1}\sum_{i=1}^{T-1}\mathbb{E}\left[\|\boldsymbol{w}_T - \boldsymbol{w}_i^*\|^2 - \|\boldsymbol{w}_i - \boldsymbol{w}_i^*\|^2\right]\\ =&\frac{1}{T-1}\sum_{i=1}^{T-1}\|\boldsymbol{w}_T^* - \boldsymbol{w}_i^*\|^2.\end{aligned}$$