Algorithm Design for Continual Learning in IoT Networks

Shugang Hao
Pillar of Engineering Systems and Design
Singapore University of Technology and Design
Singapore, Singapore
shugang_hao@sutd.edu.sg

Lingjie Duan
Pillar of Engineering Systems and Design
Singapore University of Technology and Design
Singapore, Singapore
lingjie_duan@sutd.edu.sg

***Abstract*—Continual learning (CL) is a new online learning technique over sequentially generated streaming data from different tasks, aiming to maintain a small forgetting loss on previously-learned tasks. Existing work focuses on reducing the forgetting loss under a given task sequence. However, if similar tasks continuously appear to the end time, the forgetting loss is still huge on prior distinct tasks. In practical IoT networks, an autonomous vehicle to sample data and learn different tasks can route and alter the order of task pattern at increased travelling cost. To our best knowledge, we are the first to study how to opportunistically route the testing object and alter the task sequence in CL. We formulate a new optimization problem and prove it NP-hard. We propose a polynomial-time algorithm to achieve approximation ratios of $\frac{3}{2}$ for underparameterized case and $\frac{3}{2}+r^{1-T}$ for overparameterized case, respectively. Simulation results verify our algorithm's close-to-optimum performance.**

***Index Terms*—Continual learning, IoT network, task ordering, approximation algorithm design**

I. Introduction

Continual learning (CL) is a new online learning technique over sequentially generated streaming data from different tasks and aims to maintain a small forgetting loss on previously-learned tasks (e.g., [1]). A critical challenge in CL is catastrophic forgetting (e.g., [2]), where the agent incurs a degraded performance on previous tasks after being trained on a new one. To address this issue, many studies are proposed in the CL literature for new task learning, such as regularizing old tasks' weights (e.g., [3], [4], [5]), expanding the neural network (e.g., [6], [7], [8]) and storing data of old tasks to replay (e.g., [9], [10], [11]).

However, the existing literature focuses on reducing the forgetting loss under a given data or task sequence. If similar tasks continuously appear to the end time, the forgetting loss is still huge on the prior distinct tasks (e.g., [12]). Few studies analyze the effect of task pattern on forgetting loss, which only consider preliminary cases with a limited number of tasks or some specific task patterns (e.g., [12], [13], [14]).

In practical IoT networks, an autonomous agent to sample data and learn tasks under different contexts can actually route and alter the order of data task pattern at increased travelling cost. For example, a warehouse robot decides how to route items throughout the facility when handling online tasks like picking and packing for forgetting loss minimization (e.g., [15]). Further, Evron *et al.* (2022) in [13] presents a real-world example of autonomous vehicle training. An agent wants to learn a predictor for pedestrian detection in an autonomous vehicle, which is required to operate well in T geographically distant regions of different landscapes (e.g., city, forest, desert). He actively determines how to go through the T regions for data sampling and training to maintain a good prediction performance on each region.

Nonetheless, it is challenging for an agent to determine the order of data or task pattern in the IoT network in CL. Since tasks are geographically distant, it incurs a travelling cost when moving in the IoT network (e.g., warhouse robots or autonomous vehicles). Further, the training data is generated in an online manner, which cannot be known by the agent only after his routing decision. Later we prove that the optimization problem is NP-hard, which cannot be solved optimally in a polynomial time. It is thus required to propose a good approximation algorithm for a small ratio to the optimum.

To our best knowledge, we are the first to study *how to opportunistically route the testing object and alter the task sequence in CL*. We formulate this as a new optimization problem and prove it NP-hard. We propose an algorithm to achieve approximation ratios of $\frac{3}{2}$ for the underparameterized case and $\frac{3}{2}+r^{1-T}$ for the overparameterized case in a polynomial time in task number T, respectively, where $r := 1-\frac{n}{m}$ is a parameter of feature number m and sample number n and T is the task number. Simulation results verify our algorithm's close-to-optimum performance.

II. System Model and Problem Formulation

First, we introduce our system model of continual learning in an IoT network. Then, we formulate our optimization problem and present an essential assumption for analysis.

A. System Model of Continual Learning in the IoT Network

We use the motivating example in [13] to illustrate our system model for ease of understanding. We consider an agent who wants to learn a predictor for pedestrian detection in an autonomous vehicle, which is required to perform well in T different geographical regions. Define $\boldsymbol{\tau} := (\tau_1, \cdots, \tau_T)$ as a sequential route among the T regions or tasks. At each region τ_t for $t \in [T] := \{1, \cdots, T\}$, he drives the vehicle to collect

data $(\boldsymbol{x}_{\tau_t}, \boldsymbol{y}_{\tau_t})$ (e.g., by taking images of all the pedestrians in the region) for the predictor training in the region.

We denote $\boldsymbol{x}_{\tau_t}$ as an $m \times n$ feature vector with m features (e.g., image pixel) and n samples. The agent only knows that each element $x_{\tau_t}^{(i,j)}$ of $\boldsymbol{x}_{\tau_t}$ is independent of each other for $i \in [m]$ and $j \in [n]$, which follows a normal distribution as $x_{\tau_t}^{(i,j)} \sim \mathcal{N}(0,1)$ (e.g., [13], [14]). We denote $\boldsymbol{y}_{\tau_t}$ as the $n \times 1$ output vector (e.g., score of pedestrian prediction). Following the CL literature (e.g., [13], [14]), we model that each output vector $\boldsymbol{y}_{\tau_t}$ is realized according to a linear regression model:

$$\boldsymbol{y}_{\tau_t} = \boldsymbol{x}_{\tau_t}^T \boldsymbol{w}_{\tau_t}^* + \boldsymbol{z}_{\tau_t},$$

where $\boldsymbol{w}_{\tau_t}^*$ is the $m \times n$ vector of the ground-truth model parameters (e.g., feature weights) and $\boldsymbol{z}_{\tau_t}$ is the $n \times 1$ noise vector (e.g., image background). Each element $z_{\tau_t}^{(i)}$ of $\boldsymbol{z}_{\tau_t}$ is independent of each other for $i \in [n]$ and follows a Gaussian distribution as $z_{\tau_t}^{(i)} \sim \mathcal{N}(0, \sigma^2)$ (e.g., [13], [14]).

The agent only knows the distribution of the noise and is uncertain of neither the ground-truth vector $\boldsymbol{w}_{\tau_t}^*$ nor the noise realization $\boldsymbol{z}_{\tau_t}$. In practice, an autonomous device may have low computational and memory resources, preventing it from storing data of old tasks (e.g., [15]). We then consider a memoryless setting to learn each predictor $\boldsymbol{w}_{\tau_t}$ by minimizing the training loss given the obtained data $(\boldsymbol{x}_{\tau_t}, \boldsymbol{y}_{\tau_t})$ as follows:

$$L_{\tau_t}(\boldsymbol{w}) = \frac{1}{n} ||\boldsymbol{x}_{\tau_t}^T \boldsymbol{w} - \boldsymbol{y}_{\tau_t}||_2^2. \tag{1}$$

In Sections III and IV, we analyze both the underparameterized case (i.e., $m \leq n-2$) and overparameterized case (i.e., $m \geq n+2$). The cases of $m \in \{n-1, n, n+1\}$ are undefined for each Inverse-Wishart distributed $(\boldsymbol{x}_t^T \boldsymbol{x}_t)^{-1}$, which is essential to obtain the solution $\boldsymbol{w}_{\tau_t}$ to (1) (e.g., [16]).

The agent then ships the vehicle from the region τ_t to next τ_{t+1} to sample data and train there. We denote $c_{i,j}$ as the travelling distance or cost between regions i and j, which satisfies $c_{i,j} \leq c_{i,k} + c_{k,j}$ for any $i, j, k \in [T]$ and $i \neq j \neq k$.

As the agent may still receive tasks from previously-visited regions in the future, he expects a small generalized loss of his final predictor $\boldsymbol{w}_{\tau_T}$ in the last training region τ_T from all the ground-truth model parameters $\{\boldsymbol{w}_{\tau_t}^*\}_{t=1}^T$. We thus define his forgetting loss F_T in all T regions as the average sum of the squared ℓ_2-norm distance as follows (e.g., [14]):

$$F_T(\boldsymbol{w}_{\tau_T}) = \frac{1}{T} \sum_{t=1}^{T} ||\boldsymbol{w}_{\tau_T} - \boldsymbol{w}_{\tau_t}^*||^2. \tag{2}$$

When routing in the IoT network, he incurs a travelling cost $\sum_{t=1}^{T-1} c_{\tau_t, \tau_{t+1}}$. Thus, we define his overall loss $\pi(\boldsymbol{\tau})$ as the sum of the forgetting loss in (2) and average travelling cost among the T regions as follows:

$$\pi(\boldsymbol{\tau}) = \frac{1}{T} \sum_{t=1}^{T} ||\boldsymbol{w}_{\tau_T} - \boldsymbol{w}_{\tau_t}^*||^2 + \frac{1}{T} \sum_{t=1}^{T-1} c_{\tau_t, \tau_{t+1}}. \tag{3}$$

Note that the agent cannot control any online data input (x_{τ_t}, y_{τ_t}) but the travelling order $\boldsymbol{\tau}$.

B. Problem Formulation

Based on our system model above, we are now ready to formulate an optimization problem for the agent, which involves the following two stages.

- *Stage I.* The agent determines a route $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_T)$ over the T regions for minimizing the expectation of his overall loss in (3).
- *Stage II.* In each region τ_t for $t \in [T]$, he learns a predictor $\boldsymbol{w}_{\tau_t}$ by minimizing the training loss $L_{\tau_t}(\cdot)$ in (1).

In practice, each ground-truth parameter $\boldsymbol{w}_{\tau_i}^*$ is unknown to the agent (e.g., [14]). Thus, the dissimilarity of ground-truth parameters $||\boldsymbol{w}_{\tau_i}^* - \boldsymbol{w}_{\tau_j}^*||^2$ between regions τ_i and τ_j is unknown to the agent. Further, the dissimilarity between each region τ_t's ground-truth parameter $\boldsymbol{w}_{\tau_t}^*$ and the agent's initial predictor $\boldsymbol{w}_0$ is unknown to the agent, either. Similar to [14], we assume without loss of generality in the following.

Assumption 1: The dissimilarity between ground-truth parameters $\boldsymbol{w}_{\tau_i}^*$ of region τ_i and $\boldsymbol{w}_{\tau_j}^*$ of region τ_j is upper-bounded as $||\boldsymbol{w}_{\tau_i}^* - \boldsymbol{w}_{\tau_j}^*||^2 \leq \Delta_{\tau_i, \tau_j}$ for $i \neq j$ and $i, j \in [T]$. The dissimilarity between each region τ_t's ground-truth parameter $\boldsymbol{w}_{\tau_t}^*$ and the agent's initial predictor $\boldsymbol{w}_0$ is upper-bounded as $||\boldsymbol{w}_{\tau_t}^* - \boldsymbol{w}_0||^2 \leq \Delta_{\tau_t, 0}$ for $t \in [T]$.

The agent can infer each Δ_{τ_i, τ_j} and $\Delta_{\tau_i, 0}$ for $i \neq j$, $i, j \in [T]$ based on his historical data (e.g., [14]).

III. The Agent's Expected Forgetting Loss in Closed Form and NP-Hardness

In this section, we first derive the closed-form formulation of the agent's expected forgetting loss in the underparameterized and overparameterized cases, respectively. Then, we prove that the optimization problem is NP-hard.

A. Analysis of the Underparameterized Case

In Stage II, for each region τ_t, $t \in [T]$, the agent aims to learn the predictor $\boldsymbol{w}_{\tau_t}$ for minimizing the training loss $L_{\tau_t}(\cdot)$ in (1). According to [14], in the underparameterized case of $n \geq m+2$, minimizing (1) returns a unique solution $\boldsymbol{w}_{\tau_t}$:

$$\boldsymbol{w}_{\tau_t} = (\boldsymbol{x}_{\tau_t} \boldsymbol{x}_{\tau_t}^T)^{-1} \boldsymbol{x}_{\tau_t} \boldsymbol{y}_{\tau_t}. \tag{4}$$

After substituting $\boldsymbol{w}_{\tau_t}$ in (4) into the agent's forgetting loss F_T in (2) and taking expectation over the feature vector $\boldsymbol{x}_{\tau_t}$ and the noise vector $\boldsymbol{z}_{\tau_t}$, we have the following.

Lemma 1: In the underparameterized case of $n \geq m+2$, the agent's expected forgetting loss $\mathbb{E}[F_T^u]$ is in closed form:

$$\mathbb{E}[F_T^u] = \sum_{i=1}^{T-1} \frac{||\boldsymbol{w}_{\tau_T}^* - \boldsymbol{w}_{\tau_i}^*||^2}{T} + \frac{m\sigma^2}{n-m-1}. \tag{5}$$

Substituting $\mathbb{E}[F_T^u]$ in (5) into (3), we obtain the agent's expected overall loss $\mathbb{E}[\pi_u(\boldsymbol{\tau})]$ in closed form as follows:

$$\mathbb{E}[\pi_u(\boldsymbol{\tau})] = \frac{1}{T} \sum_{i=1}^{T-1} ||\boldsymbol{w}_{\tau_T}^* - \boldsymbol{w}_{\tau_i}^*||^2 + \frac{1}{T} \sum_{t=1}^{T-1} c_{\tau_t, \tau_{t+1}} + \frac{m\sigma^2}{n-m-1}. \tag{6}$$

Since the agent just knows each Δ_{τ_i,τ_j} as the upper-bound of the dissimilarity of ground-truth parameters $||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}^*_{\tau_j}||^2$ for $i \neq j$, $i,j \in [T]$, he is only aware of an upper-bound $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ of $\mathbb{E}[\pi_u(\boldsymbol{\tau})]$ in (6) as below:

$$\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})] := \sum_{i=1}^{T-1} \frac{\Delta_{\tau_i,\tau_T}}{T} + \sum_{t=1}^{T-1} \frac{c_{\tau_t,\tau_{t+1}}}{T} + \frac{m\sigma^2}{n-m-1}. \quad (7)$$

In Stage I, he focuses on minimizing the upper-bound $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ in (7) when determining the travelling order $\boldsymbol{\tau}$:

$$\min_{\boldsymbol{\tau}=(\tau_1,\cdots,\tau_T)} \mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]. \quad (8)$$

B. Analysis of the Overparameterized Case

In Stage II, for each region τ_t, $t \in [T]$, the agent aims to learn the predictor $\boldsymbol{w}_{\tau_t}$ for minimizing the training loss $L_{\tau_t}(\cdot)$ in (1). According to [14], in the overparameterized case of $m \geq n+2$, minimizing (1) returns infinite solutions with zero loss. To preserve as much information about old tasks as possible, here we focus on the solution that has the smallest ℓ_2-norm distance with $\boldsymbol{w}_{\tau_{t-1}}$, which is the convergent point of the stochastic gradient descent method as follows:

$$\boldsymbol{w}_{\tau_t} = \boldsymbol{w}_{\tau_{t-1}} + \boldsymbol{x}_{\tau_t}(\boldsymbol{x}^T_{\tau_t}\boldsymbol{x}_{\tau_t})^{-1}(\boldsymbol{y}_{\tau_t} - \boldsymbol{x}^T_{\tau_t}\boldsymbol{w}_{\tau_{t-1}}). \quad (9)$$

Define $r{:=}1{-}\frac{n}{m}$. After substituting $\boldsymbol{w}_{\tau_t}$ in (9) into forgetting loss F_T in (2) and taking expectation over the feature vector $\boldsymbol{x}_{\tau_t}$ and the noise vector $\boldsymbol{z}_{\tau_t}$, we have the following.

Lemma 2: In the overparameterized case of $m \geq n+2$, the agent's expected forgetting loss $\mathbb{E}[F^o_T]$ is in closed form:

$$\mathbb{E}[F^o_T] = \sum_{i=1}^{T} \frac{(1-r)r^{T-i}}{T} \sum_{j=1}^{T} ||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}^*_{\tau_j}||^2 + \frac{r^T}{T}\sum_{i=1}^{T} ||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}_0||^2 + \frac{(1-r^T)m\sigma^2}{m-n-1}. \quad (10)$$

Substituting $\mathbb{E}[F^o_T]$ in (10) into (3), we obtain the agent's expected overall loss $\mathbb{E}[\pi_o(\boldsymbol{\tau})]$ in closed form as follows:

$$\mathbb{E}[\pi_o(\boldsymbol{\tau})] = \sum_{i=1}^{T} \frac{(1-r)r^{T-i}}{T} \sum_{j=1}^{T} ||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}^*_{\tau_j}||^2 + \sum_{t=1}^{T-1} \frac{c_{\tau_t,\tau_{t+1}}}{T} + \frac{r^T}{T}\sum_{i=1}^{T} ||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}_0||^2 + \frac{(1-r^T)m\sigma^2}{m-n-1}. \quad (11)$$

Since the agent just knows each Δ_{τ_i,τ_j} as the upper-bound of the dissimilarity of ground-truth parameters $||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}^*_{\tau_j}||^2$ and $\Delta_{\tau_i,0}$ as the upper-bound of the dissimilarity $||\boldsymbol{w}^*_{\tau_i} - \boldsymbol{w}_0||^2$ for $i \neq j$, $i,j \in [T]$, he is only aware of an upper-bound $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ of $\mathbb{E}[\pi_o(\boldsymbol{\tau})]$ in (11) as below:

$$\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})] = \sum_{i=1}^{T} \frac{(1-r)r^{T-i}}{T} \sum_{j=1}^{T} \Delta_{\tau_i,\tau_j} + \sum_{t=1}^{T-1} \frac{c_{\tau_t,\tau_{t+1}}}{T} + \frac{r^T}{T}\sum_{i=1}^{T} \Delta_{\tau_i,0} + \frac{(1-r^T)m\sigma^2}{m-n-1}. \quad (12)$$

In Stage I, his optimization problem is similar as (8), where he focuses on minimizing the upper-bound $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ in (12) when determining the travelling order $\boldsymbol{\tau}$.

C. NP-Hardness

One may wonder if we can efficiently solve the optimization problem of minimizing (7) or (12) in a polynomial time. Since objective functions $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ in (7) and $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ in (12) both contain the term $\sum_{t=1}^{T-1} \frac{c_{\tau_t,\tau_{t+1}}}{T}$, to solve either problem is at least as hard as solving the following problem:

$$\min_{\boldsymbol{\tau}=(\tau_1,\cdots,\tau_T)} \sum_{t=1}^{T-1} \frac{c_{\tau_t,\tau_{t+1}}}{T}. \quad (13)$$

Note that the problem in (13) is same as the classic shortest Hamiltonian path (SHP) problem in a graph $G = (V,E)$, where we denote each region τ_t as a vertex and denote each c_{τ_i,τ_j} as the weight of edge e_{τ_i,τ_j}. Since the SHP problem is known as NP-hard (e.g., [17]), we have the following.

Proposition 1: The agent's problem of minimizing either (7) in the underparameterized case or (12) in the overparameterized case is NP-hard.

Proposition 1 indicates that we cannot find the optimal solution to our problem in a polynomial time. Therefore, we are motivated to propose an efficient algorithm to find an approximation solution within a polynomial time in the region number T, guaranteed with a small approximation ratio.

IV. An Efficient Approximation Algorithm Design, Analysis and Simulation

In this section, we first present our algorithm design and its approximation ratios in the underparameterized and overparameterized cases, respectively. We then run simulations to verify our algorithm's close-to-optimum approximation.

A. Algorithm Design for the Underparameterized Case

We first present our algorithm design for the underparameterized case. According to the objective function $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ in (7), we find that only the first and the second terms of $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ depend on the travelling order $\boldsymbol{\tau}$. Further, the optimal solution to minimize the first term $\sum_{i=1}^{T-1} \frac{\Delta_{\tau_i,\tau_T}}{T}$ is to visit region T' in the end, where $T' = \arg\min_{i\in[T]} \sum_{t=1}^{T} \Delta_{i,t}$. Minimizing the second term is the same as the NP-hard SHP in (13), where we can introduce a variance of the Christofides' Algorithm (e.g., [18]) to return a $\frac{3}{2}$-approximated solution.

Based on the above analysis, we propose Algorithm 1 as an approximated algorithm to efficiently solve the problem in (8), which contains a two-layer design: first, we use Algorithm 1 to obtain a $\frac{3}{2}$-approximated solution to the problem in (13); further, we make sure that the output solution of Algorithm 1 ends with region T' to minimize the forgetting-loss part in (7).

Proposition 2: In the underparameterized case of $n \geq m+2$, the output of our Algorithm 1 incurs an approximation ratio of $\frac{3}{2}$ to the optimum of the agent's expected overall loss in (7), with a complexity order of $\mathcal{O}(T^3)$ in region number T.

B. Algorithm Design for the Overparameterized Case

We then present our algorithm design for the overparameterized case. According to the objective function $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ in (12), we find that only the first and the second terms of $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ depend on the travelling order $\boldsymbol{\tau}$.

Algorithm 1 An Approximation Algorithm

Require: Graph $G=(V,E)$, where the vertex number $|V|=T$, each edge's weight $e_{j,k}=c_{j,k}$ for all $j,k\in V$. Graph $G'=(V',E')$, where the vertex number $|V'|=T+1$, each edge's weight $e_{j,k}=c_{j,k}$ for all $j,k\in V'$, $j\neq v_0$, $k\neq v_0$, and $e_{v,v_0}=0$ for any $v\in V$ and $v\neq v_0$;

Ensure: A route $(\tau_1,\cdots,\tau_T)$.

1: Create a minimum spanning tree T of G and add the vertex v_0 to vertex v' to form a new graph T', where $v'=\arg\min_{i\in[T]}\sum_{t=1}^{T}\Delta_{i,t}$;
2: Find a minimum-weight perfect matching M in the subgraph induced in G' by O, where O is the set of vertices with odd degree in T';
3: Combine the edges of M and T' to form a connected multigraph H in which each vertex has even degree;
4: Form an Eulerian circuit in H;
5: Make the circuit found in previous step into a Hamiltonian circuit by skipping repeated vertices not connected with v_0 (shortcutting) and keeping the edge between vertices v' and v_0;
6: Remove the edges connected to the vertex v_0 to obtain a Hamiltonian path ending with vertex v'.
7: **return** The obtained Hamiltonian path.

Further, the optimal solution to minimize the first term $\sum_{i=1}^{T}\frac{(1-r)r^{T-i}}{T}\sum_{j=1}^{T}\Delta_{\tau_i,\tau_j}$ is to route in an order of

$$\boldsymbol{\tau}':=(1',\cdots,T'), \tag{14}$$

where $\sum_{t=1}^{T}\Delta_{1',t}\geq\cdots\geq\sum_{t=1}^{T}\Delta_{T',t}$. Minimizing the second term is the same as the NP-hard SHP in (13), where we can use Algorithm 1 to return a $\frac{3}{2}$-approximated solution.

Though our Algorithm 1 may not output the exact same order as $\boldsymbol{\tau}'$ in (14) for minimizing the forgetting loss, it can at least guarantee the last region same as that of $\boldsymbol{\tau}'$.

Proposition 3: In the overparameterized case of $m\geq n+2$, the output of our Algorithm 1 incurs an approximation ratio of $\frac{3}{2}+r^{1-T}$ to the optimum of the agent's expected overall loss in (7), with a complexity order of $\mathcal{O}(T^3)$ in region number T. Besides, if $m\gg n$, i.e., the feature number m is sufficiently larger than the sample number n, the ratio improves to $\frac{3}{2}$.

If the feature number m is sufficiently larger than the sample number n, we have $r{=}1-n/m{\to}1$. The first term $\sum_{i=1}^{T}\frac{(1-r)r^{T-i}}{T}\sum_{j=1}^{T}\Delta_{\tau_i,\tau_j}$ in (12) related to forgetting loss tends to zero, indicating that more features alleviate the negative impact of region dissimilarity on the forgetting loss. Therefore, the problem degenerates to (13) and our Algorithm 1 achieves a smaller approximation ratio of $\frac{3}{2}$.

According to Propositions 2 and 3, we have the following.

Corollary 1: If the region number T=2, our Algorithm 1 always returns the optimal solution to the optimization problem in both the underparameterized and overparameterized cases.

If the region number T=2, we can find that the optimal solution to minimize both $\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau})]$ in (7) and $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ in (12) is to visit region $T'=\arg\min_{i\in[T]}\sum_{t=1}^{T}\Delta_{i,t}$ in the end, which is the same as the output solution of our Algorithm 1. It then achieves the optimum.

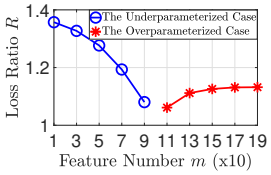

Fig. 1. The ratio R between the agent's expected overall loss $\mathbb{E}[\bar{\pi}^*]$ of our Algorithm 1 and the optimum $\mathbb{E}[\bar{\pi}^{**}]$ versus the feature number m.

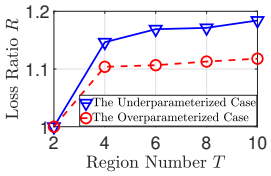

Fig. 2. The ratio R between the agent's expected overall loss $\mathbb{E}[\bar{\pi}^*]$ of our Algorithm 1 and the optimum $\mathbb{E}[\bar{\pi}^{**}]$ versus the region number T. Here we choose feature number m=80 for the underparameterized case and m=120 for the overparameterized case.

C. Simulation Results for Verification

In this section, we run simulations to show our algorithm 1's even smaller ratios than the theoretical bounds. We randomly generate each upper-bound Δ_{τ_i,τ_j}, $\Delta_{\tau_i,0}$ and each travelling cost c_{τ_i,τ_j} in the range of [1,10] without loss of generality. We fix the sample number $n=100$, change the feature number m and the region number T to verify for the underparameterized and the overparameterized cases, respectively. We define $R:=\mathbb{E}[\bar{\pi}^*]/\mathbb{E}[\bar{\pi}^{**}]$ as the ratio of the agent's expected overall loss $\mathbb{E}[\bar{\pi}^*]$ of our Algorithm 1 to the optimum $\mathbb{E}[\bar{\pi}^{**}]$, where $R{\geq}1$.

Figure 1 shows the loss ratio R versus the feature number m. Since our Algorithm 1 mainly focuses on approximating the optimal order for minimizing the travelling cost, the loss ratio R may not be monotonic in m. Nevertheless, we find that R in Figure 1 is still smaller than $\frac{3}{2}$ of Propositions 2 and 3, which implies our Algorithm 1's good approximation.

Figure 2 shows the loss ratio R versus the region number T in the underparameterized and overparameterized cases, respectively. As T increases, it is more difficult to approximate the optimal travelling order since the set of feasible order increases in a factorial way. Still, we find that the approximation ratio R in Figure 1 is smaller than $\frac{3}{2}$ of Propositions 2 and 3, which implies our Algorithm 1's good approximation.

V. Conclusion

In this paper, we study how to opportunistically route the testing object and alter the task sequence in CL. We formulate it as a new optimization problem and prove it NP-hard. We propose an algorithm to achieve approximation ratios of $\frac{3}{2}$ for the underparameterized case and $\frac{3}{2}+r^{1-T}$ for the overparameterized case in a polynomial time in task number T, respectively. Simulation results verify our algorithm's close-to-optimum performance.

REFERENCES

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

[2] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[3] W. Wang, Y. Hu, Q. Chen, and Y. Zhang, "Task difficulty aware parameter allocation & regularization for lifelong learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7776–7785.

[4] X. Zhao, H. Wang, W. Huang, and W. Lin, "A statistical theory of regularization-based continual learning," *arXiv preprint arXiv:2406.06213*, 2024.

[5] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[6] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *arXiv preprint arXiv:1708.01547*, 2017.

[7] X. Zhang, D. Song, Y. Chen, and D. Tao, "Topology-aware embedding memory for continual learning on expanding networks," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4326–4337.

[8] H. Li, S. Lin, L. Duan, Y. Liang, and N. B. Shroff, "Theory on mixture-of-experts in continual learning," *arXiv preprint arXiv:2406.16437*, 2024.

[9] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] H. Shi and H. Wang, "A unified approach to domain incremental learning with memory: Theory and algorithm," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11] X. Nie, S. Xu, X. Liu, G. Meng, C. Huo, and S. Xiang, "Bilateral memory consolidation for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 026–16 035.

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

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

[14] S. Lin, P. Ju, Y. Liang, and N. Shroff, "Theory on forgetting and generalization of continual learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 21 078–21 100.

[15] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, "Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p. 9, 2022.

[16] R. J. Muirhead, *Aspects of multivariate statistical theory*. John Wiley & Sons, 2009.

[17] M. Sevaux, K. Sörensen *et al.*, "Hamiltonian paths in large clustered routing problems," in *Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME*, vol. 8, 2008, pp. 411–417.

[18] R. van Bevern and V. A. Slugina, "A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem," *Historia Mathematica*, vol. 53, pp. 118–127, 2020.

APPENDIX

A. Proof of Lemma 1

Please refer to Appendix D.8 in reference [14].

B. Proof of Lemma 2

Please refer to Appendix D.3 in reference [14].

C. Proof of Proposition 2

Before we formally prove Proposition 2, let us introduce a useful lemma with a formal proof first.

Lemma 3: The output of Algorithm 1 incurs an approximation ratio of $\frac{3}{2}$ to the optimum of the problem in (13).

Proof. First, we have the weight $W(T')$ of the graph T' is same as that $W(T)$ of the minimum spanning tree (MST) T due to $e_{v,v_0}=0$. Denote OPT as the weight of the optimal Hamiltonian path or solution to the SHP in (13), we then have

$$W(T')=W(T)\leq \text{OPT}$$

since the weight of a Hamiltonian path (as a spanning tree) is no larger than that of a MST.

Further, according to [18], the weight $W(M)$ of a minimum-weight perfect matching M in the subgraph induced in G' by O is no larger than half of that of the optimal Hamiltonian cycle of the graph G'. Since $e_{v,v_0}=0$ for all $v\in V'$ and $v\neq v_0$ in the graph G', we have the weight of the optimal Hamiltonian cycle of the graph G' is the same as that of the optimal Hamiltonian path of the graph G, which implies

$$W(M)\leq \frac{1}{2}\text{OPT}.$$

After shortcutting the multigraph H (obtained by combining the edges of M and T') to obtain a Hamiltonian circuit C, we have the weight of C can only be decreased due to the triangle inequality $c_{ij}\leq c_{ik}+c_{kj}$ for $i,j,k\in V$ and $i\neq j\neq k$, i.e.,

$$W(C)\leq W(T')+W(M)\leq \frac{3}{2}\text{OPT}.$$

After removing the edges connected to the vertex v_0 in the Hamiltonian circuit C to obtain a Hamiltonian path P ending with vertex v', we have the weight of P is the same as that of C due to $e_{v,v_0}=0$ for all $v\in V'$ and $v\neq v_0$. Finally, we have

$$W(P)=W(C)\leq \frac{3}{2}\text{OPT}.$$

We then finish the proof. □

We then formally prove Proposition 2. Denote $\boldsymbol{\tau}^*$ as the route returned by Algorithm 1 and $\boldsymbol{\tau}^{**}$ as the optimal route. We have

$$\frac{\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau}^*)]}{\mathbb{E}[\bar{\pi}_u(\boldsymbol{\tau}^{**})]}=\frac{\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^*})]+C_T(\boldsymbol{\tau}^*)}{\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^{**}})]+C_T(\boldsymbol{\tau}^{**})}<\frac{C_T(\boldsymbol{\tau}^*)}{C_T(\boldsymbol{\tau}^{**})}\leq\frac{3}{2}.$$

The first inequality holds because $\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^*})]$ is minimized by ending with region T' and thus $\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^*})]\leq\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^{**}})]$. The second inequality holds due to Lemma 3. Therefore, we have Algorithm 1 returns a solution of an approximation ratio $\frac{3}{2}$ to the optimum. The complexity order $\mathcal{O}(T^3)$ comes from finding the minimum perfect matching M, which has been proved in [18]. We then finish the proof.

D. Proof of Proposition 3

We first prove the ratio in the general overparameterized case $m\geq n+2$. Denote $\boldsymbol{\tau}^*$ as the training order returned by Algorithm 1 and $\boldsymbol{\tau}^{**}$ as the optimal training order. We have

$$\begin{aligned}
\frac{\mathbb{E}[\bar{\pi}(\boldsymbol{\tau}^*)]}{\mathbb{E}[\bar{\pi}(\boldsymbol{\tau}^{**})]}&=\frac{\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^*})]+\mathbb{E}[C_T(\boldsymbol{\tau}^*)]}{\mathbb{E}[F_T(\boldsymbol{w}_{\tau_T^{**}})]+\mathbb{E}[C_T(\boldsymbol{\tau}^{**})]}\\
&<\frac{\frac{1}{T}\sum_{i=1}^{T}(1-r)r^{T-i}\sum_{j=1}^{T}\Delta_{\tau_i^*,\tau_j^*}+\mathbb{E}[C_T(\boldsymbol{\tau}^*)]}{\frac{1}{T}\sum_{i=1}^{T}(1-r)r^{T-i}\sum_{j=1}^{T}\Delta_{\tau_i^{**},\tau_j^{**}}+\mathbb{E}[C_T(\boldsymbol{\tau}^{**})]}\\
&<\frac{\frac{1}{T}\sum_{i=1}^{T}(1-r)\sum_{j=1}^{T}\Delta_{\tau_i^*,\tau_j^*}+\mathbb{E}[C_T(\boldsymbol{\tau}^*)]}{\frac{1}{T}\sum_{i=1}^{T}(1-r)r^{T-1}\sum_{j=1}^{T}\Delta_{\tau_i^{**},\tau_j^{**}}+\mathbb{E}[C_T(\boldsymbol{\tau}^{**})]}\\
&<\frac{\frac{1}{T}\sum_{i=1}^{T}(1-r)\sum_{j=1}^{T}\Delta_{\tau_i^*,\tau_j^*}}{\frac{1}{T}\sum_{i=1}^{T}(1-r)r^{T-1}\sum_{j=1}^{T}\Delta_{\tau_i^{**},\tau_j^{**}}}+\frac{\mathbb{E}[C_T(\boldsymbol{\tau}^*)]}{\mathbb{E}[C_T(\boldsymbol{\tau}^{**})]}\\
&=r^{1-T}+\frac{\mathbb{E}[C_T(\boldsymbol{\tau}^*)]}{\mathbb{E}[C_T(\boldsymbol{\tau}^{**})]}\leq r^{1-T}+\frac{3}{2}.
\end{aligned}$$

The first inequality holds because we subtract the common term $\frac{r^T}{T}\sum_{i=1}^{T}\Delta_{\tau_i,0}+\frac{(1-r^T)m\sigma^2}{m-n-1}$ which is independent of the order. The second inequality holds due to $r^{T-i}\in[r^{T-1},1]$. The third inequality holds due to $\frac{a+b}{c+d}<\frac{a}{c}+\frac{b}{d}$ if $a,b,c,d>0$. The second equality holds since the term $\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{T}\Delta_{\tau_i,\tau_j}$ has the same value regardless of training order. The last inequality holds due to Lemma 3. Therefore, we have Algorithm 1 returns a solution with $o(\frac{3}{2}+r^{1-T})$ approximation ratio to the optimum. The complexity order $\mathcal{O}(T^3)$ comes from finding the minimum perfect matching M, which has been proved in [18].

If $m\gg n$, we have $r=1-n/m\to 1$ and the objective function $\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]$ in (12) becomes

$$\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau})]=\sum_{t=1}^{T-1}\frac{c_{\tau_t,\tau_{t+1}}}{T}+\frac{1}{T}\sum_{i=1}^{T}\Delta_{\tau_i,0}.$$

Denote $\boldsymbol{\tau}^*$ as the route returned by Algorithm 1 and $\boldsymbol{\tau}^{**}$ as the optimal route. We have

$$\frac{\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau}^*)]}{\mathbb{E}[\bar{\pi}_o(\boldsymbol{\tau}^{**})]}=\frac{\frac{1}{T}\sum_{i=1}^{T}\Delta_{\tau_i^*,0}+C_T(\boldsymbol{\tau}^*)}{\frac{1}{T}\sum_{i=1}^{T}\Delta_{\tau_i^{**},0}+C_T(\boldsymbol{\tau}^{**})}<\frac{C_T(\boldsymbol{\tau}^*)}{C_T(\boldsymbol{\tau}^{**})}\leq\frac{3}{2}.$$

The first inequality holds because we subtract the common term $\frac{1}{T}\sum_{i=1}^{T}\Delta_{\tau_i,0}$ which is independent of the travelling order. The second inequality holds due to Lemma 3. Therefore, we have Algorithm 1 returns a solution of an approximation ratio $\frac{3}{2}$ to the optimum. We then finish the proof.