Latency-aware Online Continual Learning for Non-Stationary Data Streams

Haibo Liu, Da Huo, Zhenzhe Zheng and Fan Wu
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
Emails:{liuhaibo, sjtuhuoda, zhengzhenzhe}@sjtu.edu.cn, fwu@cs.sjtu.edu.cn

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044660

***Abstract*—Online continual learning (CL) is beneficial for learning incrementally from continuous data streams without forgetting previously learned knowledge. However, current online CL approaches have overlooked the time cost of online data collection and model adaptation, resulting in a high-latency service response, especially in high-velocity non-stationary data streams. In this work, we aim to realize latency-aware online CL for non-stationary data streams, and propose a two-stage time-scale optimization for online data collection and model adaptation. In the first stage with uncertain data arrivals, we propose an optimal stopping algorithm with a logarithmic regret bound to make an irrevocable decision on when to stop data collection. To minimize the training time of model adaptation for stability-plasticity trade-off in the second stage, we introduce a bidirectional data selection algorithm with a logarithmic approximation, to greedily determine which samples to select from both newly collected data and the previous ones. Extensive evaluations demonstrate that our proposed approach consistently outperforms the-state-of-art solutions, improving the accuracy by 16.8% on average and reducing the latency by up to 6.2 times.**

***Index Terms*—Online Continual Learning, Latency, Stability and Plasticity, Non-stationary Data Streams**

I. Introduction

There is an increasing demand to provide machine learning models as a service to support time-sensitive intelligent applications, *e.g.*, real-time person identification through surveillance cameras [1], object detection with unmanned aerial vehicles (UAV) [2] and traffic light detection for autonomous vehicles [3]. The prevalence of non-stationary data streams in these online applications poses great challenge for its service performance guarantee. To enhance performance for online service provisioning, the machine learning model should be upgraded to adapt to new arriving data samples. Consequently, it is essential and imperative to learn incrementally from non-stationary data streams under a latency constraint.

Online CL has gained increasing attention for its ability to learn incrementally from data streams, by enabling frequent model retraining to adapt to new arriving data samples, while not forgetting the previously learned knowledge [4]–[6]. A majority of online CL approaches have been proposed to overcome catastrophic forgetting, including experience replay based on gradient diversity [7], shapley values [8], and mutual information [5]. However, existing works only focused on how to realize stability-plasticity trade-off during the learning process, while often overlooking the low-latency requirements of online service provisioning. As demonstrated in computationally budgeted CL [9], the existing CL approaches, including distillation [10], sampling [7], FC layers correction [11] and model expansions [12], fail to have good model performance in a latency-constraint setting. Therefore, it is challenging to realize online CL for non-stationary data streams with performance guarantee under a strict latency constraint of response time.

The latency of learning incrementally from non-stationary data streams comprises two components: the waiting time due to online data collection and the training time required for model adaptation. However, the existing online CL approaches neglect the waiting time of online data collection, and regard data samples arriving at once for model adaptation, without exploring the impact of data collection in online CL. Although some works take into account the training time of model adaptation, they still relied on outdated models for online service provisioning, resulting in subpar model performance [13]. Furthermore, most approaches neglected the conflict between the waiting time of online data collection and the training time of model adaptation, leading to high-latency service responses, especially for high-velocity data streams. To achieve latency-aware online CL, it is crucial to determine both the waiting time for online data collection and the training time for model adaptation, thereby balancing model performance and response latency.

In this work, we focus on the latency-aware online CL for non-stationary data streams, and formulate it as a two-stage time-scale optimization problem. In the first stage, we need to determine an appropriate-sized time window to perform online data collection, ensuring that the collected data samples are effective enough for the subsequent model adaptation, while minimizing the waiting time of online data collection. In the second stage, we aim to conduct efficient and effective data selection both on the newly collected data samples and the historical stored data samples, to make the trade-off between model accuracy and training time. However, solving this two-stage time-scale optimization problem presents two significant challenges. First, without the information of future arrival data, it is hard to make the optimal decision on the online data

This work was supported in part by National Key R&D Program of China (No. 2023YFB4502400), in part by China NSF grant No. 62322206, 62132018, 62025204, U2268204, 62272307, 62372296. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government. Zhenzhe Zheng is the corresponding author.

collection. Furthermore, due to the non-stationarity of data streams, new arriving data samples evolve over time, making it harder to predict their underlying distribution, and decide when to stop the data collection. Second, performing data selection on both newly collected data samples and previously stored data samples to achieve a stability-plasticity trade-off is challenging under the strict time constraints of model adaptation. Compared to selecting data samples for replay solely from previously stored dataset, the bidirectional data selection is more challenging due to the conflict between the stability on previous data samples and the plasticity on new ones. Moreover, the coupling between the model performance and its training time with respect to the quality and quantity of data samples makes it difficult to identify an efficient coreset.

To address the first challenge, we formulate the problem as an optimal stopping problem with a Markov process to capture the non-stationarity of data streams. At each time step, an irrevocable decision must be conducted on whether to continue collecting data or to stop. Continuously interacting with non-stationary data streams can learn the shifting pattern of the non-stationarity of data streams, but would increase the waiting time. By using information entropy to measure the diversity of newly collected data samples as the reward feedback for online data collection, we introduce an optimal stopping algorithm with a logarithmic regret upper bound. As for the second challenge, to achieve stability-plasticity trade-off, we minimize the gradient discrepancy computed from the selected data samples and the full ones to improve its plasticity, and control the gradient angle computed from the data samples selected from new dataset and the ones selected from previous dataset to prevent catastrophic forgetting. To avoid the extensive computation of gradients, we approximate the differences between gradients by bounding them with the differences among data samples. We further divide the data selection on newly collected data samples and previously accumulated ones separately, and propose a bidirectional data selection strategy to start data selection on the new dataset, and then followed by the previous ones. For theoretical analysis, we leverage the submodular optimization and utilize a straightforward greedy approach to find the efficient solution with a logarithmic approximation.

Our contributions in this work are summarized as follows.

- In this work, we focus on the latency of learning incrementally from non-stationary data streams, by considering the time cost of online data collection and model adaptation, and propose a two-stage time-scale optimization approach for latency-aware online CL.
- In the first stage with uncertain data arrivals, we utilize information entropy to characterize data diversity, and propose an optimal stopping algorithm with a logarithmic regret bound to make an irrevocable decision on when to stop online data collection.
- To minimize the training time of model adaptation for stability-plasticity trade-off in the second stage, we introduce a bidirectional data selection algorithm with a logarithmic approximation, to greedily determine which samples to select from both newly collected data and the previous ones.
- Extensive evaluations demonstrate that our proposed approach consistently outperforms the-state-of-art solutions, improving the accuracy by 16.8% on average and reducing the response latency by up to 6.2 times.

II. Preliminaries

A. Non-Stationary Data Streams

In non-stationary data streams, data samples arrive in sequence over time steps T. At every time step $t \in \{1, 2, ..., T\}$, the current data samples $(x_t, y_t) \sim \mathcal{D}_t$ consist of an input feature $x_t \in \mathcal{X}$ and its corresponding label $y_t \in \mathcal{Y}$, where $\mathcal{D}_t$ represents the data distribution. In order to characterize the distribution change of data streams, we adopt a Markov process $\{\mathcal{D}_t, t \geq 1\}$ with state space $\mathcal{D}$ and transition probability matrix P, which can be described as follows:

$$P^t(d_i, d_j) = P(\mathcal{D}_{t+1} = d_i | \mathcal{D}_t = d_j), \forall d_i, d_j \in \mathcal{D}, \quad (1)$$

where $P^t(d_i, d_j)$ is the conditional probability of changing from the distribution d_j at time step t to the distribution d_i at time step $t+1$. Therefore, given the initial data distribution $\mathcal{D}_1$ and the transition probabilities $\{P^k\}_{k=1}^t$, we can describe the distribution of data samples at next time step:

$$\mathcal{D}_{t+1} = \mathcal{D}_t P^t = \mathcal{D}_1 \prod_{k=1}^{t} P^k. \quad (2)$$

We leverage the conditional probability of the new data distribution at the next time step to characterize the non-stationarity of data streams. If the probability of the new data distribution at the next time step is high, it indicates a high level of non-stationarity, otherwise with a low-level one. Consequently, we can characterize the non-stationarity of data streams as follows:

$$P^t(\mathcal{D}_{t+1} \notin \{D_k\}_{k=1}^t | \{D_k\}_{k=1}^t) = \alpha, \quad (3)$$

where α is a hyperparameter over the level of non-stationarity. Control over α enables exploring continuous data streams with different levels of non-stationarity.

B. Latency-aware Online CL

We adopt latency-aware online CL paradigm that employs a pipeline for online data collection and model adaptation with multiple rounds, as illustrated in Figure 1. In the i-th round, the latency L_i of online CL comprises two parts: the waiting time w_i of the i-th online data collection and the training time τ_i of the i-th model adaptation, *e.g.*, $L_i = w_i + \tau_i$. After collecting enough new arriving data samples, we assume that the i-th model adaptation starts at the time step s_i. The waiting time of the i-th online data collection is defined as the interval between the $(i-1)$-th model adaptation and the i-th one, *e.g.*, $w_i = s_i - s_{i-1}$. The training time of the i-th model adaptation is modeled linearly as $f(\cdot)$ with respect to the number of data samples $\mathcal{K}_i$ utilized for model adaptation [14], *i.e.*,

$$\tau_i = f(\mathcal{K}_i) = \kappa(|\mathcal{K}_i|) + \zeta, \quad (4)$$

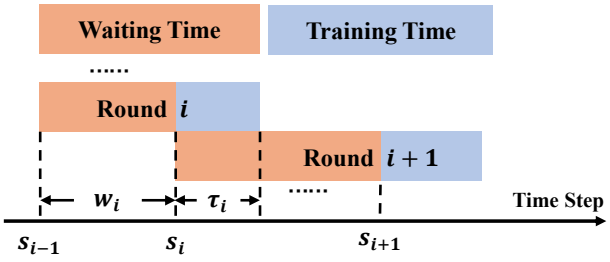

Fig. 1. The illustration of latency-aware online CL.

where κ and ζ are algorithm-specific parameters.

To effectively learn from non-stationary data streams, online CL needs to perform continuous model adaptation to new data samples. Let $\mathcal{B}_i$ be the newly collected data samples between the $(i-1)$-th and i-th model adaptation, *e.g.*, $\mathcal{B}_i = \{(x_t, y_t)|s_{i-1} \leq t \leq s_i\}$, and $\mathcal{C}_i = \{\mathcal{B}_j\}_{j=1}^{i-1}$ be the previously accumulated ones. In the i-th model adaptation, we seek to learn a model θ_i that maps the input $x_t \in \mathcal{X}$ to the label $y_t \in \mathcal{Y}$. The objective of online CL is to adapt to the current data distribution while not forgetting the previous knowledge:

$$\begin{aligned}\mathcal{L}_i(\mathcal{B}_i, \mathcal{C}_i) =& \frac{1}{|\mathcal{B}_i|} \sum_{(x_t,y_t)\in\mathcal{B}_i} l(x_t, y_t; \theta_i) \\ &+ \frac{\lambda}{|\mathcal{C}_i|} \sum_{(x_t,y_t)\in\mathcal{C}_i} l(x_t, y_t; \theta_i),\end{aligned} \tag{5}$$

where l indicates any standard loss function (*e.g.*, cross-entropy loss), and λ is the regularization weight.

C. Problem Formulation

To realize low-latency online CL for non-stationary data streams with performance guarantee, it is essential to determine the time step s_i to start model adaptation and a dataset $\mathcal{K}_i \subseteq \mathcal{B}_i \cup \mathcal{C}_i$ as training data for the trade-off between the model performance $\mathcal{L}_i$ and response latency L_i. Therefore, the optimization problem for latency-aware online CL can be formulated as follows:

$$\min_{s_i, \mathcal{K}_i} \mathcal{L}_i(\mathcal{B}_i, \mathcal{C}_i) + \gamma L_i(s_i, \mathcal{K}_i), \tag{6}$$

where γ is the regularization weight, and L_i denotes the function of response latency.

Considering the conflicts between data collection and model adaptation, we first decompose the optimization problem in Eq. (6), and then reformulate it as a two-stage time-scale optimization problem. In the first stage, we want to maximize the information diversity of the collected data samples, while minimizing its waiting time. Thus, the time optimization problem in the first stage can be formulated as:

$$s_i^* = arg \max_{s_i} H(\{(x_t, y_t)|s_{i-1} < t \leq s_i\}) - \mu w_i, \tag{7}$$

where $H(\{(x_t, y_t)|s_{i-1} < t \leq s_i\})$ is the information entropy of the collected data samples, and μ is the regularization weight. In the second stage, we would minimize the loss function and the training time of model adaptation by selecting the data samples $\mathcal{K}_i$. To adapt to new data samples without catastrophic forgetting, we divide the data samples $\mathcal{K}_i$ into two parts: the coreset selected from newly collected data samples $\mathcal{M}_i \subseteq \mathcal{B}_i$ and the one from previously accumulated data samples $\mathcal{N}_i \subseteq \mathcal{C}_i$, *e.g.*, $\mathcal{K}_i = \mathcal{M}_i \cup \mathcal{N}_i$. Consequently, the optimization problem in the second stage can be described as:

$$\mathcal{M}_i^*, \mathcal{N}_i^* = arg \min_{\mathcal{M}_i, \mathcal{N}_i} \mathcal{L}_i(\mathcal{B}_i, \mathcal{C}_i) + \gamma \tau_i. \tag{8}$$

III. Two-Stage Time-Scale Optimization

In this section, we first develop an optimal stopping algorithm to address the online data collection problem in the first stage. Subsequently, we introduce a bidirectional data selection algorithm to achieve efficient data selection in the second stage. We then present a comprehensive two-stage time-scale optimization algorithm across multiple rounds, and provide its convergence analysis along with performance guarantees.

A. Stage I: Online Data Collection

In the first stage, we need to determine a stopping time s_i for the i-th online data collection. Our goal for this stage is to maximize the accuracy of the updated model on both the newly emerging data samples and existing ones, while minimizing the waiting time required for the data collection process. Considering that the accuracy of the updated model cannot be obtained in advance, we leverage information entropy as a substitute metric to evaluate the efficiency of the collected data samples. Higher information entropy implies that the collected data samples are more valuable for model retraining to improve its plasticity. Specifically, we model the online data collection process as a discrete-time, infinite-horizon optimal stopping problem. At each time step $t \in \{s_{i-1}, \cdots, T\}$, we can choose to stop the data collection process, or continue with the increasing waiting time. We explicitly compute the information entropy of the collected data samples with its label categories. The category state space is $C = \{c_1, c_2, \cdots, c_n\}$ and its labels $y \in C$. Consequently, we aim to maximize the information entropy of the collected data samples minus its waiting time by re-writing the objective in Eq. (7) as:

$$\max_{s_i} \Big(- \sum_{c\in\mathcal{C}} p_{Y_{s_i}}(c) \log p_{Y_{s_i}}(c) - \mu \cdot (s_i - s_{i-1}) \Big), \tag{9}$$

where $Y_{s_i} = \{y_{s_{i-1}}, \cdots, y_{s_i}\}$ is the label set of the collected dataset, and $p_{Y_{s_i}}(c)$ means its probability of each label.

Considering that the arriving data sample adheres to a Markov process, we can utilize the Markov Decision Process to solve this optimal stopping problem. To maximize Eq. (9), we treat the reward brought by each newly collected data y_t as the increase in information entropy along with the time penalty. The decision to continue collecting new data samples yields the corresponding reward, while the decision to stop has no impact on the reward. However, the entropy increment of a newly collected data sample cannot be determined solely by itself; it necessitates the entire dataset Y_{s_i}. By viewing Y_{s_i} as the corresponding state of the new data sample at next time

step, directly calculating its entropy increment and reward may lead to an excessively large sample space. For efficient model retraining, we approximate the entropy increment by using the l nearest samples to the new ones, instead of the entire dataset Y_{s_i}. With the assumption that the distribution of the l data samples mirrors that of Y_{s_i}, the sign of the entropy increment is identical. Formally, we define the state space, action space, and rewards of this optimization problem as follows:

- State $\mathbb{S}$: $S_t = \{y_{t-l+1}, y_{t-l+2}, \cdots, y_t\} \in \mathbb{S}$.
- Action $\mathbb{A}$: $\mathbb{A} = \{\text{stop}, \text{continue}\}$.
- Reward R: $R(S_t) = -\mu + H(S_t) - H(S_{t-1})$.

Due to the data distribution shift conforming to the Markov property, the state transition here also adheres to a Markov process. Given the state transition probability $P_{S_iS_j}$ from the state S_i to the another one S_j, we can calculate the maximum expected future reward $Q^*(S_t)$ obtained by taking the action to continue in state S_t,

$$Q^*(S_t) = R(S_t) + \sum_{S_{t+1} \in \mathbb{S}} P_{S_tS_{t+1}} \max\{Q^*(S_{t+1}), 0\}. \quad (10)$$

Correspondingly, adopting the action to stop yields a future reward that is perpetually 0. Thus, the optimal policy $\pi : \mathbb{S} \to \mathbb{A}$ can be defined by

$$\pi^*(S_t) = \begin{cases} \text{stop}, & \text{if } Q^*(S_t) \le 0 \\ \text{continue}, & \text{otherwise.} \end{cases} \quad (11)$$

We learn the unknown state transition probabilities from the non-stationary environment of data streams. According to the Bellman equation, the widely utilized approach for updating the Q-function is

$$Q(S_t) \leftarrow (1-\beta)Q(S_t) + \beta\left[R(S_t) + \phi V(S_{t+1})\right], \quad (12)$$

where $V(S_{t+1}) = \max\{Q(S_{t+1}), 0\}$.

We solve the optimal stopping problem through Algorithm 1. We begin by continuously collecting l data samples to form S_l (Line 2). Subsequently, for each newly collected data samples, we determine whether to stop using the Q-value and the optimal policy π^* (Line 4). If we decide to stop ($a = 0$), the stopping time is returned. Otherwise, we compute the subsequent state along with the entropy increment and the associated waiting time introduced by the new sample collection (Line 6-7). Ultimately, we update the Q-value based on Eq. (12). To reduce the regret of the optimal stopping problem algorithm, we implement the UCB-Hoeffding algorithm to refine the updates made to the Q-values [15]–[17]. In contrast to Eq. (12), an additional term b has been incorporated into the update of the Q-function (Line 10), which serves as a confidence bonus reflecting the algorithm's confidence level regarding the currently explored Q value. Specifically, according to the UCB algorithm, we define the bonus term b:

$$b = \frac{4\sqrt{2}}{1-\phi}\sqrt{\frac{\mathcal{H}\log(2\mathcal{S}T(N(S_t)+1)(N(S_t)+2))}{N(S_t)}} \quad (13)$$

Algorithm 1: Optimal Stopping Algorithm

Input: State space size $\mathcal{S}$, the length of data streams T, discount factor ϕ, learning rate β.
Output: The stopping time t.

1 Initialize $Q(S), \widehat{Q}(S) \leftarrow \frac{1}{1-\phi}$, $N(S) \leftarrow 0$;
2 $S_l = \{y_1, y_2, \cdots, y_l\}$, $a \leftarrow 1$, $t \leftarrow l$;
3 **while** $a \neq 0$ **do**
4 $a \leftarrow \pi^*(S_t)$;
5 **if** $a = 1$ **then**
6 $S_{t+1} \leftarrow S_t \setminus \{y_{t-l}\} \cup \{y_{t+1}\}$;
7 $R(S_{t+1}) \leftarrow H(S_{t+1}) - H(S_t) - \mu$;
8 $N(S_t) \leftarrow N(S_t) + 1$;
9 $\widehat{V}(S_{t+1}) \leftarrow \max\{\widehat{Q}(S_{t+1}), 0\}$;
10 $Q(S_t) \leftarrow (1-\beta)Q(S_t)$
11 $+\beta\left[R(S_t) + b + \phi\widehat{V}(S_{t+1})\right]$;
12 $\widehat{Q}(S_t) \leftarrow \min\left\{\widehat{Q}(S_t), Q(S_t)\right\}$;
13 $t \leftarrow t+1$;
14 **return** t.

where $\mathcal{H} \leftarrow \frac{\ln(2/(1-\phi)\delta)}{\ln(1/\phi)}$ is a constant with hyperparameters δ, and the learning rate $\beta \leftarrow \frac{\mathcal{H}+1}{\mathcal{H}+N(S_t)}$ diminishes progressively with the accumulation of learning epochs $N(S_t)$. For Algorithm 1, we can prove that the upper bound of its regret is less than $\mathcal{O}\left(\log\frac{2\mathcal{S}T}{(1-\phi)}\right)$ [15].

B. Stage II: Bidirectional Data Selection

In the second stage, given the newly collected dataset $\mathcal{B}_i$ and the previous one $\mathcal{C}_i$, we aim to solve the optimization problem presented in Eq. (8) for the optimal $\mathcal{M}_i^* \subseteq \mathcal{B}_i$ and $\mathcal{N}_i^* \subseteq \mathcal{C}_i$, to minimize the training time of the i-th model adaptation, *e.g.*, τ_i, while adapting to new arriving data samples without forgetting previous ones, *e.g.*, $\mathcal{L}_i$. The i-th model adaptation with stochastic gradient descent can be described as follows:

$$\begin{cases} \theta_i = \theta_{i-1} - \eta(\gamma_i + \upsilon_i), \\ \gamma_i = \frac{1}{|\mathcal{M}_i|}\sum_{(x_t,y_t)\in\mathcal{M}_i} \nabla l(x_t, y_t; \theta_{i-1}), \\ \upsilon_i = \frac{1}{|\mathcal{N}_i|}\sum_{(x_t,y_t)\in\mathcal{N}_i} \nabla l(x_t, y_t; \theta_{i-1}), \end{cases} \quad (14)$$

where γ_i represents the average gradient computed from the data samples in $\mathcal{M}_i$, and υ_i is the average gradient computed from the ones in $\mathcal{N}_i$, respectively. Similarly, we define Γ_i and Υ_i as the full average gradients for the newly collected data samples $\mathcal{B}_i$ and the previously accumulated ones $\mathcal{C}_i$, *e.g.*,

$$\begin{cases} \Gamma_i = \frac{1}{|\mathcal{B}_i|}\sum_{(x_t,y_t)\in\mathcal{B}_i} \nabla l(x_t, y_t; \theta_{i-1}), \\ \Upsilon_i = \frac{1}{|\mathcal{C}_i|}\sum_{(x_t,y_t)\in\mathcal{C}_i} \nabla l(x_t, y_t; \theta_{i-1}). \end{cases} \quad (15)$$

To maximize the generalization capability of the updated model θ_i across all accumulated data samples $\{\mathcal{B}_j\}_{j=1}^i$, it is essential to minimize the gradient discrepancy between the selected data samples and the corresponding whole ones, *e.g.*, $\|\gamma_i - \Gamma_i\|$ and $\|\upsilon_i - \Upsilon_i\|$. Additionally, to adapt to the current task while preventing catastrophic forgetting [7], it is necessary

Algorithm 2: Bidirectional Data Selection Algorithm

Input: Newly collected dataset $\mathcal{B}_i$, previously dataset $\mathcal{C}_i$, threshold ϵ, regularization weight λ.
Output: new coreset $\mathcal{M}_i$, previous coreset $\mathcal{N}_i$.

$\mathcal{M}_i \leftarrow \varnothing, \mathcal{N}_i \leftarrow \varnothing, \mathcal{I}_{\mathcal{N}_i} \leftarrow 0, \mathcal{I}_{\mathcal{M}_i} \leftarrow 0$;
$\mathcal{I}_{\mathcal{C}_i} \leftarrow \sum_{x_k \in \mathcal{C}_i} \frac{x_k}{|\mathcal{C}_i|}, \mathcal{I}_{\mathcal{B}_i} \leftarrow \sum_{x_k \in \mathcal{B}_i} \frac{x_k}{|\mathcal{B}_i|}$;
/* Current Data Selection */
while $\|\mathcal{I}_{\mathcal{M}_i} - \mathcal{I}_{\mathcal{B}_i}\| \leq \frac{\epsilon}{1+\lambda}$ **do**
  $x_{\mathcal{M}_i} \leftarrow arg\max_{x \in \mathcal{B}_i \setminus \mathcal{M}_i} \mathcal{G}(\mathcal{M}_i \cup \{x\}) - \mathcal{G}(\mathcal{M}_i)$;
  $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup \{x_{\mathcal{M}_i}\}$;
  $\mathcal{I}_{\mathcal{M}_i} \leftarrow \sum_{x_k \in \mathcal{M}_i} \frac{x_k}{|\mathcal{M}_i|}$;
/* Previous Data Selection */
while $\|\mathcal{I}_{\mathcal{N}_i} - \mathcal{I}_{\mathcal{C}_i}\| \leq \frac{\epsilon\lambda}{1+\lambda}$ **do**
  $x_{\mathcal{N}_i} \leftarrow arg\max_{x \in \mathcal{C}_i \setminus \mathcal{N}_i} \mathcal{G}(\mathcal{N}_i \cup \{x\}) - \mathcal{G}(\mathcal{N}_i)$ *s.t.* $\langle \mathcal{I}_{\mathcal{N}_i}, \mathcal{I}_{\mathcal{M}_i} \rangle \geq 0$;
  $\mathcal{N}_i \leftarrow \mathcal{N}_i \cup \{x_{\mathcal{N}_i}\}$;
  $\mathcal{I}_{\mathcal{N}_i} \leftarrow \sum_{x_k \in \mathcal{N}_i} \frac{x_k}{|\mathcal{N}_i|}$;
return $\mathcal{M}_i$, $\mathcal{N}_i$.

Algorithm 3: LATE

Input: Initialized model θ_0, state space size $\mathcal{S}$, the length of stream T, discount factor ϕ, learning rate β, threshold ϵ, regularization weight λ.
Output: i-th updated model θ_i.

$i \leftarrow 1$, $s_0 \leftarrow 0$;
for $t = 1 : T$ **do**
  /* Stage I: Data Collection */
  $s_i \leftarrow$ Algorithm 1($\mathcal{S}$, T, ϕ, β);
  if $t == s_i$ **then**
    $\mathcal{B}_i \leftarrow \{x_k, y_k\}_{k=s_{i-1}}^{s_i}$;
    $\mathcal{C}_i \leftarrow \{\mathcal{B}_k\}_{k=1}^{s_{i-1}}$;
    /* Stage II: Data Selection */
    $\mathcal{M}_i, \mathcal{N}_i \leftarrow$ Algorithm 2($\mathcal{B}_i, \mathcal{C}_i$, ϵ, λ);
    /* Parallel: Model Adaptation */
    $\gamma_i = \frac{1}{|\mathcal{M}_i|} \sum_{(x;y) \in \mathcal{M}_i} \nabla l(x, y; \theta_{i-1})$;
    $\upsilon_i = \frac{1}{|\mathcal{N}_i|} \sum_{(x;y) \in \mathcal{N}_i} \nabla l(x, y; \theta_{i-1})$;
    $\theta_i = \theta_{i-1} - \eta(\gamma_i + \upsilon_i)$;
    $i \leftarrow i + 1$;
return θ_i .

to control the gradient angle computed from new selected data samples and previous selected ones, *e.g.*, $\langle \gamma_i, \upsilon_i \rangle \geq 0$. Consequently, the objective function of the bidirectional data selection problem in Eq. (8) can be reformulated as follows:

$$\begin{aligned} &\mathcal{M}_i^*, \mathcal{N}_i^* = arg\min |\mathcal{M}_i| + |\mathcal{N}_i|, \\ &s.t.\ \|\gamma_i - \Gamma_i\| + \lambda \|\upsilon_i - \Upsilon_i\| \leq \epsilon, \langle \gamma_i, \upsilon_i \rangle \geq 0, \end{aligned} \tag{16}$$

where the error ϵ ensures that the sampling gradients closely approximate the full gradients.

Solving the optimization problem in Eq. (16) is challenging and costly due to the large combination space and the extensive computation required for full gradients across all accumulated data samples. To alleviate the extensive computation of gradients, we approximate the differences between gradients derived from data samples using the techniques in [18], [19]:

$$\|\nabla l(x_i, y_i; \theta) - \nabla l(x_j, y_j; \theta)\| \leq const. \|x_i - x_j\|, \forall i, j. \tag{17}$$

In this way, we can bound the difference between the gradients γ_i and the full gradients Γ_i by the distance between the sampling data samples $\mathcal{M}_i$ and the full collected data samples $\mathcal{B}_i$. Similarly, we can compute the similarity between the data samples $\mathcal{M}_i$ and the data samples $\mathcal{N}_i$ to bound the product of gradients from these two sampled datasets, *e.g.*, $\langle \gamma_i, \upsilon_i \rangle$. Consequently, we can reformulate the constraint conditions in Eq. (16) as follows:

$$\begin{cases} \|\sum_{x_k \in \mathcal{M}_i} \frac{x_k}{|\mathcal{M}_i|} - \sum_{x_k \in \mathcal{B}_i} \frac{x_k}{|\mathcal{B}_i|}\| \leq \frac{\epsilon}{1+\lambda}\ , \\ \|\sum_{x_k \in \mathcal{N}_i} \frac{x_k}{|\mathcal{N}_i|} - \sum_{x_k \in \mathcal{C}_i} \frac{x_k}{|\mathcal{C}_i|}\| \leq \frac{\epsilon\lambda}{1+\lambda}\ , \\ \langle \sum_{x_k \in \mathcal{N}_i} \frac{x_k}{|\mathcal{N}_i|}, \sum_{x_k \in \mathcal{M}_i} \frac{x_k}{|\mathcal{M}_i|} \rangle \geq 0. \end{cases} \tag{18}$$

By applying triangular inequality, we can obtain an upper bound for the approximation to the first constraint condition in Eq. (18), and define the function $\mathcal{G}(\mathcal{M}_i)$ as follows:

$$\begin{aligned} \|\sum_{x_k \in \mathcal{M}_i} \frac{x_k}{|\mathcal{M}_i|} - \sum_{x_l \in \mathcal{B}_i} \frac{x_l}{|\mathcal{B}_i|}\| &\leq \sum_{x_l \in \mathcal{B}_i} \|\frac{x_l}{|\mathcal{B}_i|} - \sum_{x_k \in \mathcal{M}_i} \frac{x_k}{|\mathcal{B}_i|}\|, \\ &\leq \sum_{x_l \in \mathcal{B}_i} \max_{x_k \in \mathcal{M}_i} \|\frac{x_k}{|\mathcal{M}_i|} - \frac{x_l}{|\mathcal{B}_i|}\| \triangleq \mathcal{G}(\mathcal{M}_i). \end{aligned} \tag{19}$$

As for the optimization problem in Eq. (16), we can reduce it as a set cover problem, which is NP-hard, and employ a straightforward greedy approach to find the efficient approximate solution. Specifically, given $\mathcal{M}_i$ and $\mathcal{B}_i$, we greedily select the data sample in $\mathcal{B}_i$ but not in $\mathcal{M}_i$ that yields the highest gain:

$$\max_{x \in \mathcal{B}_i \setminus \mathcal{M}_i} \mathcal{G}(\mathcal{M}_i \cup \{x\}) - \mathcal{G}(\mathcal{M}_i). \tag{20}$$

Incrementally select the data samples using this greedy approach until the selected dataset $\mathcal{M}_i$ satisfies the corresponding constraint condition. Similarly, we then utilize the same strategy to obtain $\mathcal{N}_i$. As summarized in Algorithm 2, we realize the bidirectional data selection by first performing data selection on the new dataset (Line 3-6), and then followed by the selection on the previous ones (Line 7-10).

Before presenting the performance analysis of Algorithm 2, we first introduce the definition of submodular function, and then prove that the function $\mathcal{G}$ in Eq. (19) is a submodular function, and Algorithm 2 provides a solution with a logarithmic approximation in Theorem 1.

Definition 1. *A set function $f : 2^N \rightarrow R$ defined on a finite set N is called a submodular function, if for every $A \subseteq B \subseteq N$ and every element $x \in N \setminus B$, the following inequality holds:*

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B). \tag{21}$$

Theorem 1. *The function $\mathcal{G}$ is a submodular function, and the bidirectional data selection algorithm with greedy approach provides a solution with a logarithmic approximation.*

C. Algorithm Design and Analysis

In the previous sections, we introduced two algorithms for online data collection and model adaptation within a single round. Expanding upon these two algorithms, we have now developed a multi-round optimization algorithm, referred to as low-latency continual learning (LATE). LATE not only executes online data collection and selection sequentially for a single round, but also concurrently performs the data collection of the next round and the model adaptation of the current round, both of which are inherently time-sensitive tasks.

We present LATE, the online two-stage time-scale optimization algorithm shown in Algorithm 3. Initially, LATE initializes the variables i and s_0 (Line 1), where i represents the index of the current round of model adaptation, and s_0 denotes the initial time point. At each time step t, LATE employs Algorithm 1 to determine whether to stop online data collection (Line 3). If Algorithm 1 continues proceeding with data collection, it runs without any return value. However, if Algorithm 1 stops online data collection, it yields the optimal stopping time s_i. Depending on the decision in the first stage, LATE initiates bidirectional data selection using Algorithm 2, aiming to identify the optimal training samples $\mathcal{M}_i$ and $\mathcal{N}_i$ from the newly gathered data samples $\mathcal{B}_i$ and the previously accumulated ones $\mathcal{C}_i$ (Line 4-7). Following in this manner, LATE simultaneously performs the model adaptation and initiates the next round of data collection to reduce the time cost. Specifically, LATE calculates gradients for the collected and selected datasets $\mathcal{M}_i$ and $\mathcal{N}_i$, aggregates these gradients, and updates the model parameters from θ_{i-1} to θ_i (Line 8-11). Concurrently, LATE restarts the subsequent data collection stage and reactivates Algorithm 1 to identify the optimal time point s_{i+1} for online data collection in the next round, thus forming a continuous cycle of data collection and model adaptation (back to Line 3).

Next, we provide an analysis of the convergence of LATE. Before giving the corresponding convergence analysis, we provide the following assumptions.

- (Lipschitz gradient) For any model parameters θ_i and θ_j, there exist a constant $L > 0$ that makes the loss gradient $\nabla\mathcal{L}(\theta_j)$ satisfies $\|\nabla\mathcal{L}(\theta_i) - \nabla\mathcal{L}(\theta_j)\| \leq L\|\theta_i - \theta_j\|$.
- (Bound gradient variance) There exist a constant σ, such that $E[\|\gamma_i - \Gamma_i\|] \leq \sigma, E[\|\upsilon_i - \Upsilon_i\|] \leq \sigma, \forall i$.
- (Bound loss) There exist the supremum of loss gap $\triangle_{\mathcal{L}}$ between θ_1 and θ^*, *i.e.*, $\triangle_{\mathcal{L}} = \sup \mathcal{L}(\theta_1) - \mathcal{L}(\theta^*)$.

According to the above assumptions, we provide the convergence analysis of Algorithm 3 in Theorem 2. It demonstrates that the minimum value of the gradient produced by LATE is bounded by the sampled gradients and converges over time.

Theorem 2. *Considering the model parameter θ, collected dataset $\mathcal{B}_i$, learning rate η, bound gradient variance σ and bounded loss $\triangle_{\mathcal{L}}$, the iterates of LATE satisfy:*

$$\frac{1}{T}\sum_{t=1}^{T} E\|\nabla\mathcal{L}(\theta_t)\|^2 \leq \frac{L\eta}{2(1-\frac{L\eta}{2})}\sigma^2 + \frac{1}{T\eta(1-\frac{L\eta}{2})}(\sum_{t=1}^{T}(\Theta_t + \Phi_t) + \triangle_{\mathcal{L}}) \tag{22}$$

where $\Theta_t = \frac{L\eta^2}{2}\|\upsilon_t\|^2 + (L\eta^2 - \eta)\langle\upsilon_t, \gamma_t\rangle$, $e_t = \nabla\mathcal{L}(\theta_{t-1}) - \gamma_t$, and $\Phi_t = (\eta - L\eta^2)\langle\nabla\mathcal{L}(\theta_{t-1}), e_t\rangle - \eta\langle\upsilon_t, e_t\rangle$.

IV. Performance Evaluation

In this section, we first describe the experimental setup, and then report the experimental results with overall performance and key parameter analysis.

A. Experiment Setting

1) **Datasets and Models**: we utilize two different image datasets: CIFAR100 and Tiny-ImageNet to testify the effectiveness of our method. As for machine learning models, we select two lightweight models: MobileNet V2 and ResNet18. As for these models, the learning rate and batch size of model retraining are 0.01 and 32, respectively. Moreover, we set the decay factor and momentum to their default values of 0. As for data streams, we consider class-incremental CL settings [20], [21] by dividing the whole classes of each dataset into different data groups with the length of data streams T, and configure the arrival interval for data streams to be 1s.

2) **Baselines**: We compare our proposed LATE with the following classical CL methods:

- MIR, Maximally Interfered Retrieval, a memory retrieval method that retrieves memory samples that suffer from an increase in loss given the estimated parameters update based on the current task [22].
- GDUMB, Greedy Sampler and Dumb Learner, greedily stores samples from data streams, and trains a model from scratch using samples only in the memory [23].
- AGEM, Averaged Gradient Episodic Memory, a memory-based method that utilizes the samples in the memory buffer to constrain the parameter updates [24].

3) **Metrics**: We evaluate the performance of LATE by using the following two metrics:

- Average Accuracy: We use $\triangle f(x_t, y_t; \theta)$ to denote the prediction accuracy of predictor θ where x_t is the input feature, y_t is the truth label and t is the time step. The average accuracy at the end of data streams can be measured as follows:

$$\mathcal{A}(T) = \frac{1}{T}\sum_{t=1}^{T}\triangle f((x_t, y_t; \theta). \tag{23}$$

- Response Latency: We compute the waiting time w_i of the i-th online data collection and the training time τ_i of the i-the model adaptation, and add these two time cost as its response latency:

$$L = \frac{1}{T}\sum_{t=1}^{T} L_t = w_t + \tau_t. \tag{24}$$

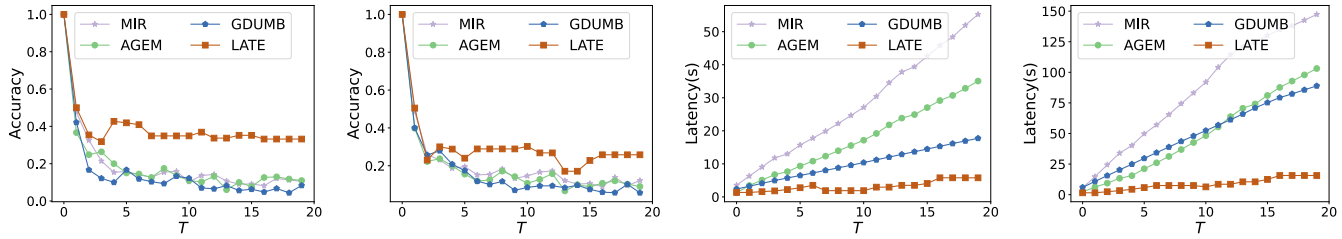

Fig. 2. The performance comparison of four CL approaches: (a) Accuray of MobileNet V2 on CIFAR100; (b) Accuracy of ResNet18 on Tiny-ImageNet; (c) Latency of MobileNet V2 on CIFAR100; (d) Latency of ResNet18 on Tiny-ImageNet.

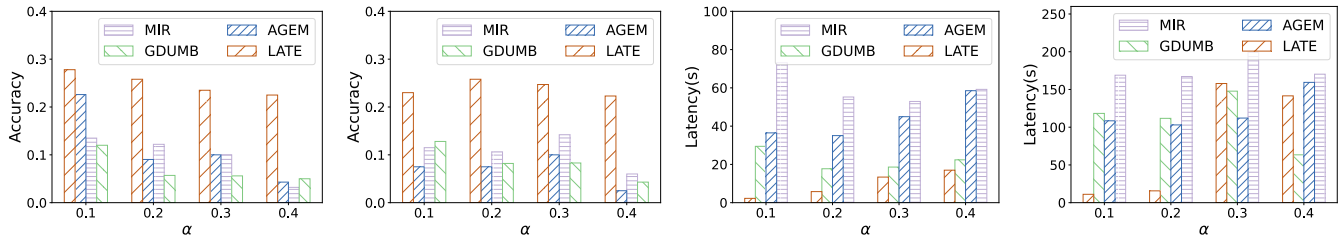

Fig. 3. The impact of α: (a) Accuray of MobileNet V2 on CIFAR100; (b) Accuracy of ResNet18 on Tiny-ImageNet; (c) Latency of MobileNet V2 on CIFAR100; (d) Latency of ResNet18 on Tiny-ImageNet.

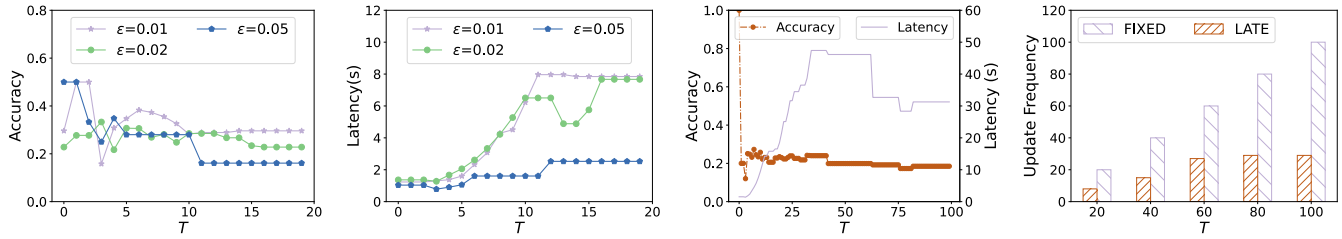

Fig. 4. (a) The impact of ϵ on accuracy; (b) The impact of ϵ on latency; (c) The impact of T on accuracy and latency; (d) The frequency of model adaptation.

B. Overall Performance

Figure 2 illustrates the average accuracy and response latency of four different CL methods with time steps $T = 20$. Figure 2(a) depicts the average accuracy of MobileNet V2 on the dataset CIFAR100, showing that the average accuracy of these four CL methods decreases over time steps. At time steps $T = 20$, it is evident that LATE outperforms other CL methods with a 21% performance improvement. The main reason for this phenomenon is that LATE performs online data collection and selects valuable samples from both the newly accumulated dataset and previous ones. The proposed LATE enhances its plasticity without catastrophic forgetting, making it more efficient compared to other CL methods. To further validate this result in a more persuasive manner, we conducted additional experiments using ResNet18 on the Tiny-ImageNet dataset. As shown in Figure 2(b), the results confirm the same conclusion: LATE outperforms the other CL methods with an 11% improvement in average accuracy. Figure 2(c) plots the response latency of MobileNet V2 on the dataset CIFAR100, demonstrating that the response latency of these four CL methods increases over time steps. At time step $T = 20$, it is observed that MIR has the highest response latency at approximately 55 seconds, while LATE has the lowest response latency at around 8 seconds. Consequently, without frequent model adaptation to avoid data congestion, LATE outperforms the other CL methods with an average acceleration of 2.5 times. Similarly, as shown in Figure 2(d), we further validate the response latency of these four CL methods by using ResNet18 on the Tiny-ImageNet dataset. Although the response latency on ResNet18 is relatively high compared to the lightweight model MobileNet V2, the acceleration in response latency is up to 7.5 times between LATE and the other CL methods, further demonstrating the superiority and robustness of LATE.

C. Key Parameter Analysis

Impact of α. In figure 3, we explore the impact of non-stationarity of data streams on the average accuracy and response latency of these four CL methods. Figure 3(a) plots the average accuracy of MobileNet V2 on the CIFAR-100

dataset with four different values of α. It is evident that the average accuracy of these four CL methods is higher in scenarios with low-degree non-stationarity data streams. Regardless of whether the non-stationarity is high or low, LATE consistently demonstrates superior model performance compared to the other CL methods. Similarly, we can obtain the same conclusion by further using ResNet18 on the Tiny-ImageNet dataset shown in Figure 3(b). Figure 3(c) plots the response latency of MobileNet V2 on the dataset CIFAR100 with different value of α, and we find that the response latency of LATE increases with the value of α, whereas the response time of other methods does not show a direct correlation with the non-stationarity of data streams. This indicates that only the LATE method optimizes for different non-stationarity of data streams, demonstrating higher adaptability compared to the other three CL methods. Moreover, LATE has shortest response time comparing to other CL methods, and can be further validated in Figure 3(d) with more experiment by utilizing ResNet18 on the Tiny-ImageNet.

Impact of ϵ. We use parameter ϵ to control the amount of data collection and selection. The larger the parameter value, the less data needs to be collected and selected. In Figure 4, we investigate the impact of ϵ on the average accuracy and response latency. Figure 4(a) shows the average accuracy and response latency of MobileNet V2 on the CIFAR-100 dataset with different values of ϵ. It is evident that both the average accuracy and response latency of LATE increase as the collected and selected data samples increase. Therefore, LATE with ϵ achieves the highest accuracy compared to the other settings. Additionally, Figure 4(b) compares the response latency of model adaptation with different values of ϵ, showing that the response latency increases with the amount of data. It is clear that LATE exhibits the smallest response latency.

Impact of T. We further investigate the impact of the length of time steps on the average accuracy and response latency of these four CL methods. Figure 4(c) shows the average accuracy and response latency of MobileNet V2 on the CIFAR-100 dataset over time steps $T = 100$. It is evident that both the average accuracy and response latency of LATE fluctuate initially but stabilize as T increases. Additionally, Figure 4(d) compares the frequency of model adaptation between LATE and the other CL methods, revealing that LATE requires fewer adaptations. While the frequency of model adaptation for the other CL methods increases linearly with time steps, LATE shows a relatively slower rate of increase.

V. Related Works

Non-stationary Data Streams. Massive data streams are continuously collected from ubiquitous end devices, and required immediate processing to satisfy the low latency requirements of many real-world applications [1]–[3], [25]–[27]. In traditional continual learning scenarios, data arrives in batches, with each batch consisting of i.i.d. samples and available in large quantities. However, in the context of non-stationary data streams, data arrives gradually over time, and the data distribution also evolves with time [28], [29]. For instance, a UAV equipped with a high-resolution camera captures images for object detection on the fly, and the categories of these images may undergo significant changes due to the non-stationarity of environment [2]. To tackle the non-stationarity of continuous data streams, machine learning model should be retrained or replaced to adapt to new arriving data [4]–[6]. To learn online from data streams with frequent data distribution change, a system for enabling learning and prediction at same time has been proposed, where a novel objective function synchronizes the latent space with the continually evolving prototypes [28]. However, existing works only center on how to realize efficient model training or model selection, while often overlooking the low-latency requirements of online service provisioning. Consequently, it is essential and imperative to learn incrementally from non-stationary data streams under the latency constraint.

Online Continual Learning. Online CL is becoming a mainstream paradigm to learn incrementally from continuous data streams without forgetting previously learned knowledge. The current practices to overcome catastrophic forgetting can be identified into three main families of approaches: Regularization methods add extra terms to the loss function to limit the model's capacity, thereby preventing the model from overfitting to new learning tasks [30]–[32]. Replay methods reintroduce old data during the training process to help the model recall old knowledge [33]–[35]. Expansion methods increase the model's capacity to learn new tasks while maintaining access to old knowledge [36]–[38]. Although in an online manner, current online CL methods are still unable to support online service provisioning, as most of them overlook the conflict between the waiting time of online data collection and the training time of model adaptation. There also exist a few works in online CL to focus on the training time of model adaptation. In order to realize inference queries at any time, the authors design a new memory management scheme and learning rate scheduling strategy to adapt in online, task-free, class-incremental of blurry task streams [39]. To evaluate current CL methods, a new real-time evaluation in online CL has been proposed to consider the training delay and fast change in data distribution [13]. Although these methods can perform model inference without any delay, the performance of online service provisioning is subpar due to the reuse of an older model, especially for slow-training CL methods and high-velocity data streams with fast data distribution change.

Data Selection. Data selection refers to the process of choosing a coreset from a large dataset, ensuring that this selected coreset is similar to the original one. There exist various approaches to obtain a coreset from a large dataset. Importance sampling amplifies the loss/gradients of significant samples based on influence functions [40]. Recently, a bilevel optimization framework has been proposed to incorporate cardinality constraints for coreset selection [41]. To evaluate the importance of data, a large amount of research works mainly focus on metrics like loss [42], gradient norm [43], [44], uncertainty [45], [46], shapely value [47] and representativeness [48]. In continual learning, data selection is often used

to choose the data that needs to be replayed. They select replay samples based on loss increment [22], gradient diversity [7], shapely values [8], mutual information [5] and other deliberately designed algorithms [23], [49], [50]. However, existing method is extremely limited in practice and inapplicable in online service provisioning settings due to the excessive computational cost incurred during model adaptation. In contrast, our proposed approach utilizes gradients as the criterion for data selection, and simultaneously perform data selection on both historical and new data to reduce the time cost of online data collection and model adaptation, which is beneficial for learning incrementally from non-stationary data streams.

VI. Conclusion

In this work, we focused on the latency of learning incrementally from non-stationary data streams, and have proposed a two stage time-scale optimization approach to realize low-latency online CL. In the first online stage with uncertain data arrivals, we have designed an optimal stopping algorithm with a logarithmic regret bound to make an irrevocable decision on when to perform model adaptation. To reduce the training time of model adaptation in the second stage, we introduced a greedy sample selection algorithm to determine samples to be used from both new accumulated data and previous ones, to improve its plasticity without catastrophic forgetting. Extensive evaluations demonstrate that our proposed approach consistently outperforms the-state-of-art solutions, improving the accuracy by 16.8% on average and reducing the response latency by up to 6.2 times.

Appendix

A. Proof of Theorem 1

Proof. We define the function $\mathcal{G}$ in Eq. (19), and prove that this function is a submodular one as following. First, given $A \subseteq \mathcal{M}_i \subseteq \mathcal{B}_i$ and every element $x \in \mathcal{B}_i \backslash \mathcal{M}_i$, we can have:

$$\begin{aligned}\mathcal{G}(\mathcal{M}_i \cup \{x\}) - \mathcal{G}(\mathcal{M}_i) = \sum_{x_l \in \mathcal{U}} \| \frac{x}{|\mathcal{M}_i|+1} - \frac{x_l}{|\mathcal{B}_i|} \| \\ - \sum_{x_l \in \mathcal{U}} \max_{x_k \in \mathcal{M}_i} \| \frac{x_k}{|\mathcal{M}_i|+1} - \frac{x_l}{|\mathcal{B}_i|} \|,\end{aligned} \tag{25}$$

where U denotes the the set of data samples with the change of the maximum value. Similarly, we can further obtain:

$$\begin{aligned}\mathcal{G}(A \cup \{x\}) - \mathcal{G}(A) = \sum_{x_l \in \mathcal{U}} \| \frac{x}{|A|+1} - \frac{x_l}{|\mathcal{B}_i|} \| - \\ \sum_{x_l \in \mathcal{U}} \max_{x_k \in \mathcal{M}_i} \| \frac{x_k}{|\mathcal{A}|+1} - \frac{x_l}{|\mathcal{B}_i|} \| + \sum_{x_l \in \mathcal{O}} \| \frac{x}{|A|+1} - \frac{x_l}{|\mathcal{B}_i|} \| \\ - \sum_{x_l \in \mathcal{O}} \max_{x_k \in \mathcal{M}_i} \| \frac{x_k}{|\mathcal{A}|+1} - \frac{x_l}{|\mathcal{B}_i|} \|.\end{aligned} \tag{26}$$

Due to $A \subseteq \mathcal{M}_i \subseteq \mathcal{B}_i$, the set of data samples with the change of the maximum value is in A. Besides, there also exists the other set $\mathcal{O}$ in A but not in $\mathcal{M}_i$. Thus, we can conclude it with $\mathcal{G}(A \cup \{x\}) - \mathcal{G}(A) \geq \mathcal{G}(\mathcal{M}_i \cup \{x\}) - \mathcal{G}(\mathcal{M}_i)$. As for the logarithmic approximation, the proof is similar to [51] and we omit the details due to the space limit □

B. Proof of Theorem 2

Proof. Let $\gamma_t = \frac{1}{|\mathcal{N}_t|} \sum_{(x_i,y_i) \in \mathcal{N}_t} \nabla l(x_i, y_i; \theta_{i-1})$ and $\upsilon_t = \frac{1}{|\mathcal{M}_t|} \sum_{(x_i,y_i) \in \mathcal{M}_t} \nabla l(x_i, y_i; \theta_{t-1})$. Moreover, we define $\nabla \mathcal{L}(\theta_{t-1}) = \frac{1}{|\mathcal{B}_t|} \sum_{(x_i,y_i) \in \mathcal{B}_t} \nabla l(x_i, y_i; \theta)$ and $e_t = \nabla \mathcal{L}(\theta_{t-1}) - \gamma_t$. In this way, we can have that $\mathcal{L}(\theta_t) \leq$

$$\begin{aligned}&\mathcal{L}(\theta_{t-1}) + < \nabla \mathcal{L}(\theta_{t-1}), \theta_t - \theta_{t-1} > + \frac{L}{2} \|\theta_t - \theta_{t-1}\|^2 \\ &= \mathcal{L}(\theta_{t-1}) - \eta < \nabla \mathcal{L}(\theta_{t-1}), \upsilon_t + \gamma_t > + \frac{L\eta^2}{2} \|\gamma_t + \upsilon_t\|^2, \\ &= \mathcal{L}(\theta_{t-1}) - \eta < \nabla \mathcal{L}(\theta_{t-1}), \upsilon_t + e_t > \\ &+ \frac{L\eta^2}{2} \|\nabla \mathcal{L}(\theta_{t-1})\|^2 + \frac{L\eta^2}{2} \|e_t + \upsilon_t\|^2 \\ &+ \frac{L\eta^2}{2} \langle \nabla \mathcal{L}(\theta_{t-1}), e_t + \upsilon_t \|^2 \rangle - \eta \|\nabla \mathcal{L}(\theta_{t-1})\|^2, \\ &= \mathcal{L}(\theta_{t-1}) + (\frac{L\eta^2}{2} - \eta) \|\nabla \mathcal{L}(\theta_{t-1})\|^2 + \frac{L\eta^2}{2} \|\upsilon_t\|^2 \\ &+ \frac{L\eta^2}{2} \|e_t\|^2 + (L\eta^2 - \eta) \langle \upsilon_t, \gamma_t \rangle - \eta \langle \upsilon_t, e_t \rangle \\ &+ (\eta - L\eta^2) \langle \nabla \mathcal{L}(\theta_{t-1}), e_t \rangle.\end{aligned} \tag{27}$$

To analyze the convergence of the updated model, we need to provide proof of gradient boundedness. For this purpose, we need to transform the inequality in Eq. (27) and rewrite it as:

$$\begin{aligned}&(\eta - \frac{L\eta^2}{2}) \|\nabla \mathcal{L}(\theta_{t-1})\|^2 \leq \mathcal{L}(\theta_{t-1}) - \mathcal{L}(\theta_t) \\ &+ \frac{L\eta^2}{2} \|\upsilon_t\|^2 + \frac{L\eta^2}{2} \|e_t\|^2 + (L\eta^2 - \eta) \langle \upsilon_t, \gamma_t \rangle \\ &- \eta \langle \upsilon_t, e_t \rangle + (\eta - L\eta^2) \langle \nabla \mathcal{L}(\theta_{t-1}), e_t \rangle.\end{aligned} \tag{28}$$

For easy to read, we denoting $\Theta_t = \frac{L\eta^2}{2} \|\upsilon_t\|^2 + (L\eta^2 - \eta) \langle \upsilon_t, \gamma_t \rangle$, $\Phi_t = (\eta - L\eta^2) \langle \nabla \mathcal{L}(\theta_{t-1}), e_t \rangle - \eta \langle \upsilon_t, e_t \rangle$. Taking the expectation over the gradients $\nabla \mathcal{L}(\theta_{t-1})$, we can reformulate and have:

$$\begin{aligned}&E \|\nabla \mathcal{L}(\theta_{t-1})\|^2 \leq \frac{L\eta}{2(1 - \frac{L\eta}{2})} \|e_t\|^2 + \\ &\frac{1}{\eta(1 - \frac{L\eta}{2})} (\mathcal{L}(\theta_{t-1}) - \mathcal{L}(\theta_t) + \Theta_t + \Phi_t).\end{aligned} \tag{29}$$

Next, we further accumulate the gradients after multiple rounds of model retraining, and eliminate intermediate terms using the boundedness property in Eq. (29). Thus, we can further get the following conclusion:

$$\begin{aligned}&\frac{1}{T} \sum_{t=1}^{T} E \|\nabla \mathcal{L}(\theta_t)\|^2 \leq \frac{1}{T} \sum_{t=1}^{T} \frac{L\eta}{2(1 - \frac{L\eta}{2})} \|e_t\|^2 \\ &+ \frac{1}{T\eta(1 - \frac{L\eta}{2})} (\mathcal{L}(\theta_1) - \mathcal{L}(\theta_T) + \sum_{t=1}^{T} (\Theta_t + \Phi_t)).\end{aligned} \tag{30}$$

□

REFERENCES

[1] B. Gaikwad and A. Karmakar, "Smart surveillance system for real-time multi-person multi-camera tracking at the edge," *Journal of Real-Time Image Processing.*, vol. 18, no. 6, pp. 1993–2007, 2021.

[2] S. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao, "Development of uav-based target tracking and recognition systems," *IEEE Transactions on Intelligent Transportation Systems.*, vol. 21, no. 8, pp. 3409–3422, 2020.

[3] Z. Ouyang, J. Niu, Y. Liu, and M. Guizani, "Deep cnn-based real-time traffic light detector for self-driving vehicles," *IEEE Transactions on Mobile Computing.*, vol. 19, no. 2, pp. 300–313, 2020.

[4] A. Chrysakis and M. Moens, "Online continual learning from imbalanced data," in *Proc. of ICML*, 2020, pp. 1952–1961.

[5] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *Proc. of ICML*, 2022, pp. 8109–8126.

[6] T. L. Hayes and C. Kanan, "Online continual learning for embedded devices," in *Proc. of CoLLAs*, 2022, pp. 744–766.

[7] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Proc. of NeurIPS*, 2019, pp. 11 816–11 825.

[8] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *Proc. of AAAI*, 2021, pp. 9630–9638.

[9] A. Prabhu, H. A. A. K. Hammoud, P. K. Dokania, P. H. S. Torr, S. Lim, B. Ghanem, and A. Bibi, "Computationally budgeted continual learning: What does matter?" in *Proc. of CVPR*, 2023, pp. 3698–3707.

[10] X. Li, S. Wang, J. Sun, and Z. Xu, "Variational data-free knowledge distillation for continual learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12 618–12 634, 2023.

[11] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. of CVPR*, 2019, pp. 831–839.

[12] R. Ramesh and P. Chaudhari, "Model zoo: A growing brain that learns continually," in *Proc. of ICLR*, 2022, pp. 1–29.

[13] Y. Ghunaim, A. Bibi, K. Alhamoud, M. Alfarra, H. A. A. K. Hammoud, A. Prabhu, P. H. S. Torr, and B. Ghanem, "Real-time evaluation in online continual learning: A new hope," in *Proc. of CVPR*, 2023, pp. 11 888–11 897.

[14] H. Tian, M. Yu, and W. Wang, "Continuum: A platform for cost-aware, low-latency continual learning," in *Proc. of SoCC*, 2018, pp. 26–40.

[15] K. Yang, L. Yang, and S. Du, "Q-learning with logarithmic regret," in *Proc. of AISTATIS*, 2021, pp. 1576–1584.

[16] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is q-learning provably efficient?" in *Proc. of NeurIPS*, 2018, pp. 4868–4878.

[17] Y. Wang, K. Dong, X. Chen, and L. Wang, "Q-learning with UCB exploration is sample efficient for infinite-horizon MDP," in *Proc. of ICLR*, 2020, pp. 1–20.

[18] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *Proc. of ICML*, 2020, pp. 6950–6960.

[19] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams, "Variance reduced stochastic gradient descent with neighbors," in *Proc. of NeurIPS*, 2015, pp. 2305–2313.

[20] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *Proc. of AAAI*, 2021, pp. 9630–9638.

[21] G. Kim, C. Xiao, T. Konishi, Z. Ke, and B. Liu, "A theoretical study on solving continual learning," in *Proc. of NeurIPS.*, 2022, pp. 5065–5079.

[22] R. Aljundi, L. Caccia, E. Belilovsky, M. Caccia, M. Lin, L. Charlin, and T. Tuytelaars, "Online continual learning with maximally interfered retrieval," *CoRR*, vol. abs/1908.04742, 2019.

[23] A. Prabhu, P. H. S. Torr, and P. K. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *Proc. of ECCV*, 2020, pp. 524–540.

[24] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. of ICLR*, 2019, pp. 1–20.

[25] Y. Zhuang, Z. Zheng, F. Wu, and G. Chen, "Litemoe: Customizing on-device LLM serving via proxy submodel tuning," in *Proc. of SenSys*, 2024, pp. 521–534.

[26] C. Gong, Z. Zheng, F. Wu, X. Jia, and G. Chen, "Delta: A cloud-assisted data enrichment framework for on-device continual learning," in *Proc. of MobiCom*, 2024, pp. 1408–1423.

[27] H. Liu, J. Lu, X. Wang, C. Wang, R. Jia, and M. Li, "Fedup: Bridging fairness and efficiency in cross-silo federated learning," *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3672–3684, 2024.

[28] C. Fahy, S. Yang, and M. Gongora, "Scarcity of labels in non-stationary data streams: A survey," *ACM Computing Surveys*, vol. 55, no. 2, pp. 40:1–40:39, 2023.

[29] M. D. Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *Proc. of ICCV*, 2021, pp. 8230–8239.

[30] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, and K. Milan, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016.

[31] S. Lee, J. Kim, J. Jun, J. Ha, and B. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. of NeurIPS*, 2017, pp. 4652–4662.

[32] H. Ahn, S. Cha, D. Lee, and T. Moon, "Uncertainty-based continual learning with adaptive regularization," in *Proc. of NeurIPS*, 2019, pp. 4394–4404.

[33] C. D. Kim, J. Jeong, S. Moon, and G. Kim, "Continual learning on noisy data streams via self-purified replay," in *Proc. of ICCV*, 2021, pp. 517–527.

[34] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *Proc. of ICLR*, 2019, pp. 1–31.

[35] L. Kumari, S. Wang, T. Zhou, and J. A. Bilmes, "Retrospective adversarial replay for continual learning," in *Proc. of NeurIPS*, 2022, pp. 28 530–28 544.

[36] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *Proc. of ICML*, 2019, pp. 3925–3934.

[37] S. C. Y. Hung, C. Tu, C. Wu, C. Chen, Y. Chan, and C. Chen, "Compacting, picking and growing for unforgetting continual learning," in *Proc. of NeurIPS*, 2019, pp. 13 647–13 657.

[38] J. Yoon, S. Kim, E. Yang, and S. J. Hwang, "Scalable and order-robust continual learning with additive parameter decomposition," in *Proc. of ICLR*, 2020, pp. 1–15.

[39] H. Koh, D. Kim, J. Ha, and J. Choi, "Online continual learning on class incremental blurry task configuration with anytime inference," in *Proc. of ICLR*, 2022, pp. 1–21.

[40] S. Sinha, J. Song, A. Garg, and S. Ermon, "Experience replay with likelihood-free importance weights," in *Proc. of L4DC*, 2022, pp. 110–123.

[41] Z. Borsos, M. Mutny, and A. Krause, "Coresets via bilevel optimization for continual learning and streaming," in *Proc. of NeurIPS*, 2020, pp. 14 879–14 890.

[42] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. of CVPR*, 2016, pp. 761–769.

[43] T. B. Johnson and C. Guestrin, "Training deep models faster with robust, approximate importance sampling," in *Proc. of NeurIPS*, 2018, pp. 7276–7286.

[44] C. Gong, Z. Zheng, F. Wu, Y. Shao, B. Li, and G. Chen, "To store or not? online data selection for federated learning with limited storage," in *Proc. of WWW*, 2023, pp. 3044–3055.

[45] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Proc. of NeurIPS*, 2017, pp. 1002–1012.

[46] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. of CVPR*, 2017, pp. 2840–2848.

[47] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *Proc. of ICML*, 2019, pp. 2242–2251.

[48] Y. Wang, F. Fabbri, and M. Mathioudakis, "Fair and representative subset selection from data streams," in *Proc. of WWW*, 2021, pp. 1340–1350.

[49] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *Proc. of NeurIPS*, 2020, pp. 15 920–15 930.

[50] A. Chaudhry, N. Khan, P. K. Dokania, and P. H. S. Torr, "Continual learning in low-rank orthogonal subspaces," in *Proc. of NeurIPS*, 2020, pp. 9900–9911.

[51] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.