Tackling Spatial-Temporal Data Heterogeneity for Federated Continual Learning in Edge Networks

Junyu Shi [1] **Kun Guo** [1] **Xijun Wang** [2] **Peng Yang** [3] **Howard H. Yang** [4]

Abstract

With the rapid growth of intelligent devices such as smartphones and unmanned aerial vehicles, vast amounts of sequential data are generated at the network edge, offering rich resources for edge federated continual learning. However, the continuous influx of data introduces significant spatiotemporal heterogeneity: temporally, data distribution shifts over time lead to catastrophic forgetting; spatially, non-independent and identically distributed (non-IID) data across devices hinder global model convergence. While overcoming these challenges, it is inevitable to consider the inherent constraints of edge devices, including limited computational and storage capability. To this end, we propose Spatial-Temporal Elastic Weight Consolidation (ST-EWC) method, by which each device trains a local neural network model using only its own data within the current time period, without revisiting data from other devices and previous time periods, meanwhile local models are periodically sent to a server for global model aggregation. The key point of ST-EWC is that the local model update is guided by Fisher diagonal matrices based regularization terms applied across both spatial and temporal dimension. Experimental results demonstrate that ST-EWC significantly mitigates catastrophic forgetting, accelerates convergence, and improves average accuracy, under the settings of the temporally domain-incremental and spatially non-IID PermutedMNIST and PACS datasets.

1. Introduction

With the rapid proliferation of intelligent devices such as smartphones and unmanned aerial vehicles (UAVs), massive amounts of sequential data are generated at the network edge. This necessitates the ability of these edge devices to adapt incrementally to new data, rather than retraining from scratch. Considering the limited computational and storage capabilities of edge devices, as well as the need to preserve data privacy, performing federated continual learning (FCL) in edge networks becomes necessary. For instance, the challenge of changing data distributions caused by user mobility in edge networks is addressed (Jin et al., 2022), using an incremental learning approach that updates local models over time in a federated manner. Similarly, Wu et al. (Wu et al., 2024) tackle the poor generalization of pre-trained models and the resource limitations of UAVs by combining online learning with federated learning (FL), enabling UAVs to update their models with real-time data and adapt to new environment.

The most significant issue for FCL in edge networks arises from the dynamic nature of data, which arrives in batches over time, introducing complexities related to both temporal and spatial data heterogeneity, as shown in Figure 1. **Temporal data heterogeneity** refers to shifts in data distributions over time, driven by factors such as evolving trends, seasonal variations, and changes in user behavior. These fluctuations require models to continuously adapt, as failing to do so can lead to performance degradation. A critical issue in this context is catastrophic forgetting, where models lose previously acquired knowledge when adapting to new data. **Spatial data heterogeneity**, on the other hand, stems from the non-independent and identically distributed (non-IID) nature of data across different devices, due to the fact that each device typically collects data under unique conditions. As a result, locally trained models may fail to generalize well when aggregated into a global model, leading to slower convergence and reduced accuracy. Addressing these issues requires specialized techniques that mitigate catastrophic forgetting while ensuring global model convergence, meanwhile considering the limited computational and storage resources on edge devices.

To this end, we propose a Spatial-Temporal Elastic Weight

[1] Shanghai Key Laboratory of Multidimensional Information Processing, School of Communications and Electronics Engineering, East China Normal University, Shanghai, China [2] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China [3] School of Electronic and Information Engineering, Beihang University, Beijing, China [4] Zhejiang University/University of Illinois at Urbana-Champaign Institute, Zhejiang University, Haining, China. Correspondence to: Kun Guo <kguo@cee.ecnu.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

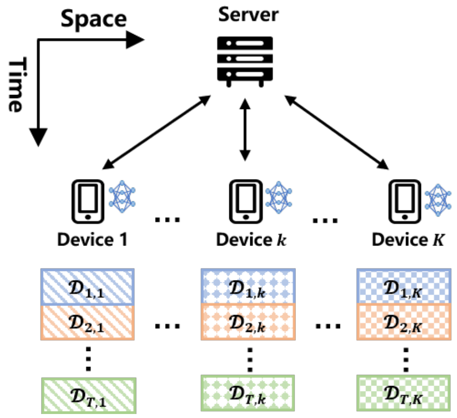


Figure 1. An illustration of spatial-temporal data distribution in edge networks.

Consolidation (ST-EWC) method to enhance the FCL in edge networks. In this approach, edge devices cooperate in the learning process with support from a centralized node, such as a server located at the network edge. Within each time period, the following process is iteratively performed until the global neural network model converges: 1) Each device updates its local model using only its own data from the current time period. To mitigating forgetting of knowledge acquired in earlier time periods, a Temporal Fisher Diagonal Matrix (T-FDM)-based regularization is applied. In parallel, to adapt to new local data while maintaining generalization to other devices' data, a Spatial Fisher Diagonal Matrix (S-FDM)-based regularization is also incorporated. 2) After completing the local model update, each device updates its S-FDM using the latest local model. 3) The updated local models and corresponding S-FDMs are sent to the server. 4) The server aggregates the local models to construct a new global model. 5) The new global model, along with other devices' updated local models and S-FDMs, is distributed back to all devices to begin the next round of local updates. Once the global model has converged within the current time period, the T-FDM is updated in a federated manner to guide model updates in future time periods. The contributions of this paper is summarized as:

- ST-EWC applies EWC constraints in both the spatial and temporal domains to guide local model updates. Specifically, the spatial EWC constraint is formulated as an S-FDM-based regularization term that identifies and preserves model parameters important for data from other devices, thereby ensuring cross-device knowledge retention. Meanwhile, the temporal EWC constraint utilizes the T-FDM to identify parameters critical to data from previous time periods, helping to retain essential knowledge from past experiences while training on current data.
- ST-EWC allows each device to train its local model using only data from the current time period, without accessing data from other devices or previous time periods. This approach significantly reduces computational and storage overhead compared to retraining on accumulated historical data, making it well-suited for resource-constrained edge devices.
- Extensive experiments on the temporally domain-incremental and spatially non-IID PermutedMNIST and PACS datasets demonstrate that ST-EWC mitigates catastrophic forgetting, accelerates convergence, and improves average accuracy by effectively balancing S-FDM-based regularization—focused on current knowledge acquisition—and T-FDM-based regularization, which mitigates catastrophic forgetting.

2. Related Work

2.1. Continual Learning

To address the learning challenge in dynamic environments where data arrives continually over time, various types of continual learning (CL) methods have been proposed to mitigate catastrophic forgetting and improve model performance across multiple time periods. These methods can be broadly categorized into three types (De Lange et al., 2019): replay-based methods, parameter isolation methods, and regularization-based methods, each of which has its own advantages and disadvantages.

Replay-based methods mitigate forgetting by storing and replaying samples from previous periods, either as raw data or pseudo-samples generated by generative models, such as iCaRL (Rebuffi et al., 2017). While effective in preserving past knowledge, these methods demand extra memory, making them less suitable for resource-limited edge networks. Parameter isolation methods allocate distinct model components (e.g., subnetworks or masks) to each period, as seen in PackNet (Mallya & Lazebnik, 2018). This kind of methods prevents interference between periods at the cost of growing neural network scale, which increases computational costs. Regularization-based methods, such as EWC (Kirkpatrick et al., 2017), avoid storing historical data by introducing temporal EWC constraints on model parameter updates. These approaches reduce computational and memory costs, but may gradually degrade performance over long sequences due to limited regularization strength. Fortunately, carefully selecting appropriate regularization terms can effectively alleviate this degradation.

In comparison, regularization-based methods—with their lower computational and memory requirements—are more suitable for edge networks. However, the aforementioned methods are typically implemented in a centralized manner, requiring raw data to be transmitted from edge devices. This leads to prohibitively high communication overhead and raises significant privacy concerns when directly applied in edge network environments.

2.2. Federated (Continual) Learning

FL (McMahan et al., 2017) offers inherent advantages for edge network environments that are resource-constrained and privacy-sensitive. However, spatial data heterogeneity incurred by the non-IID data among devices, poses significant challenges to FL. Recent studies have proposed various innovative solutions to address these challenges: The DISCO algorithm (Guo et al., 2022), which dynamically schedules device participation to tackle imbalances in data volume and variations in computational and communication capabilities across edge devices, significantly improving convergence speed and accuracy; To address the data heterogeneity in large-scale edge networks, a hierarchical personalized FL framework based on model-agnostic meta-learning has been proposed in (You et al., 2023), enhancing overall learning performance. The FedCurv algorithm, proposed in (Shoham et al., 2019), introduces a regularization term in the spatial dimension during local model updates, effectively improving the model's generalization across devices. While existing studies have extensively addressed spatial data heterogeneity, they have largely neglected the challenges posed by temporal data heterogeneity.

The combinations of FL and CL for edge artificial intelligence have been discussed in (Wang et al., 2024), enabling models to continuously adapt to dynamic data. Mainstream methods include: replay mechanism (Wei et al., 2024), (Qi et al., 2023) and its variants (Mei et al., 2024), (Usmanova et al., 2022), which store or generate a subset of past data and combine it with new data to mitigate forgetting during model updates; as well as parameter isolation methods (Yoon et al., 2021), which allocate dedicated parameter components for data from each time period and device to prevent forgetting. However, these existing methods typically require additional memory or computational resources, making them less suitable for deployment in edge networks. In contrast, our approach seeks to leverage the advantages of regularization-based method to enhance the federated continual learning on edge devices with limited resources.

3. Problem Description and Preliminaries

As illustrated in Figure 1, we consider an edge networks with K devices, in which device $k \in \mathcal{K} \triangleq \{1, ..., K\}$ sequentially collects data over time. For simplicity, we denote the data collected by device k during period $t \in \mathcal{T} \triangleq \{1, ..., T\}$ as $\mathcal{D}_{k,t}$ and $\mathcal{D}_{k,t} = \{(\boldsymbol{x}_{kj}^t, y_{kj}^t)\}_{j=1}^{n_{k,t}}$ consists of $n_{k,t}$ data samples, with $\boldsymbol{x}_{kj}^t$ and y_{kj}^t representing the input and label of the j-th sample of device k in period t. Then, the total data collected during period t is given by $\mathcal{D}_t = \bigcup_{k\in\mathcal{K}} \mathcal{D}_{k,t}$ and $\mathcal{D}_t = \{(\boldsymbol{x}_j^t, y_j^t)\}_{j=1}^{n_t}$ consists of n_t data samples, where $\boldsymbol{x}_j^i$ and y_j^i represent the input and label of the j-th data sample in period t, respectively.

We aim to find the optimal model $\boldsymbol{\theta}^t$ that performs well not only on the current period's data $\mathcal{D}_t$, but also on the previous data $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_{t-1}\}$. To achieve this, we define a global loss function to assess the performance of model $\boldsymbol{\theta}^t$, as follows:

$$L(\boldsymbol{\theta}^t, \mathcal{A}_t) = \frac{1}{|\mathcal{A}_t|} \sum_{i=1}^{t} \sum_{j=1}^{n_i} \ell(\boldsymbol{\theta}^t, \boldsymbol{x}_j^i, y_j^i), \tag{1}$$

where $\mathcal{A}_t = \bigcup_{i=1}^t \mathcal{D}_i$ represents the cumulative data up to period t and ℓ denotes a generic loss function, such as the cross-entropy loss or the mean squared error loss. It will achieve an optimal solution to update model $\boldsymbol{\theta}^t$ following (1), but along with prohibitively large amount of data to be stored on edge devices. This poses significant challenges for edge devices such as UAVs and smartphones, which typically have limited storage and computational resources.

To address this issue, we adopt the EWC method to update model $\boldsymbol{\theta}^t$ using only the current period's data, without revisiting previous data. This approach significantly reduces the computational and storage burdens on edge devices. While mitigating the forgetting of past knowledge, the EWC selectively preserves model parameters that are crucial to previous periods' data, protecting them from drastic changes during model updates. The core innovation of EWC lies in encoding parameter importance information as a regularization term in the loss function, thereby establishing an "elastic constraint" in the parameter space to safeguard learned knowledge (Aich, 2021), (Hashash et al., 2022). With the EWC, the global loss function is rewritten as:

$$\begin{aligned} L(\boldsymbol{\theta}^t, \mathcal{D}_t) = &\frac{1}{n_t} \sum_{j=1}^{n_t} \ell(\boldsymbol{\theta}^t, \boldsymbol{x}_j^t, y_j^t) \\ &+ \frac{\lambda_1}{2} \sum_{i=1}^{t-1} (\boldsymbol{\theta}^t - \boldsymbol{\theta}^{i,*})^T \boldsymbol{F}^i (\boldsymbol{\theta}^t - \boldsymbol{\theta}^{i,*}). \end{aligned} \tag{2}$$

In (2), the first term represents the loss on the current period's data, while the second term can be regarded as a temporal regularization term that protects the knowledge learned from previous data, with $\lambda_1 > 0$ as a weight factor. Additionally, $\boldsymbol{F}^i$ is the T-FDM, corresponding to the optimal model $\boldsymbol{\theta}^{i,*}$ in period i. For ease of understanding and future use, we adopt the superscript t instead of i and give the T-FDM in period t as follows:

$$\boldsymbol{F}^t = \boldsymbol{E} \odot \frac{1}{n_t} \sum_{j=1}^{n_t} \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_j^t, y_j^t)}{\partial \boldsymbol{\theta}^{t,*}} \right) \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_j^t, y_j^t)}{\partial \boldsymbol{\theta}^{t,*}} \right)^T . \tag{3}$$

Here, $\boldsymbol{E}$ is an identity matrix with the same dimensions as the T-FDM, and $\odot$ denotes element-wise matrix multiplication. Moreover, $\boldsymbol{\theta}^{t,*}$ is the optimal model for (2) minimization in period t, which can be achieved using the gradient decent method. Particularly, in the r-th gradient descent, $\boldsymbol{\theta}^t$ is updated following

$$\begin{aligned}\boldsymbol{\theta}_r^t &= \boldsymbol{\theta}_{r-1}^t - \eta \bigtriangledown L(\boldsymbol{\theta}_{r-1}^t, \mathcal{D}_t) \\ &= \boldsymbol{\theta}_{r-1}^t - \eta \left(\frac{1}{n_t} \sum_{j=1}^{n_t} \bigtriangledown \ell(\boldsymbol{\theta}_{r-1}^t, \boldsymbol{x}_j^t, y_j^t) \right. \\ &\qquad \left. + \lambda_1 \sum_{i=1}^{t-1} (\boldsymbol{\theta}_{r-1}^t - \boldsymbol{\theta}^{i,*}) \boldsymbol{F}^i \right), \end{aligned} \tag{4}$$

with learning rate η and initial model θ_0^t.

In the centralized EWC method (Hashash et al., 2022), the server executes the following procedures in period t for (2) minimization: 1) collects the data from edge device to attain $\mathcal{D}_t$; 2) updates the model following (4) to achieve the optimal solution $\boldsymbol{\theta}^{t,*}$. In this way, the catastrophic forgetting brought by the temporal data heterogeneity is mitigated effectively. However, it is often infeasible in practical edge networks to collect the data from edge devices for centralized learning, due to high communication costs and privacy concerns. Therefore, there is a need for the distributed implementation of EWC method, during which the spatial data heterogeneity among edge devices adversely affecting the learning performance has to be tackled. In this regard, we further extend the EWC method into the spatial dimension and devise ST-EWC method for edge federated continual learning. The details will be presented in the next section.

4. Our ST-EWC Method

4.1. Design Philosophy

We take period t as an example to elaborate on the distributed implementation of EWC method. The aim in period t is to achieve $\boldsymbol{\theta}^{t,*}$ using data $\mathcal{D}_t$ (distributed among K devices) to minimize the global loss in (2). Without data sharing with the server, device k instead minimizes the following local loss:

$$\begin{aligned} L_k(\boldsymbol{\theta}_k^t, \mathcal{D}_{k,t}) &= \frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \ell(\boldsymbol{\theta}_k^t, \boldsymbol{x}_{kj}^t, y_{kj}^t) \\ &\quad + \frac{\lambda_1}{2} \sum_{i=1}^{t-1} (\boldsymbol{\theta}_k^t - \boldsymbol{\theta}^{i,*})^T \boldsymbol{F}^i (\boldsymbol{\theta}_k^t - \boldsymbol{\theta}^{i,*}), \end{aligned} \tag{5}$$

where $\boldsymbol{F}^i$ for previous period i is broadcast by the server and is a known parameter for device k. Then, device k updates local model θ_k^t in the r-th gradient descent as

$$\begin{aligned} \boldsymbol{\theta}_{r,k}^t &= \boldsymbol{\theta}_{r-1,k}^t - \eta \bigtriangledown L_k(\boldsymbol{\theta}_{r-1,k}^t, \mathcal{D}_{k,t}) \\ &= \boldsymbol{\theta}_{r-1,k}^t - \eta \left(\frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \ell(\boldsymbol{\theta}_{r-1,k}^t, \boldsymbol{x}_{kj}^t, y_{kj}^t) \right. \\ &\qquad \left. + \lambda_1 \sum_{i=1}^{t-1} (\boldsymbol{\theta}_{r-1,k}^t - \boldsymbol{\theta}^{i,*}) \boldsymbol{F}^i \right). \end{aligned} \tag{6}$$

After **one-time** local gradient descent, device k then sends updated model $\boldsymbol{\theta}_{r,k}^t$ to the server. Further, the server aggregates all model updates from K edge devices as follows:

$$\begin{aligned} \boldsymbol{\theta}_r^t &= \sum_{k=1}^{K} \frac{n_{k,t}}{n_t} \boldsymbol{\theta}_{r,k}^t \\ &= \sum_{k=1}^{K} \frac{n_{k,t}}{n_t} \left(\boldsymbol{\theta}_{r-1,k}^t - \eta \frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \ell(\boldsymbol{\theta}_{r-1,k}^t, \boldsymbol{x}_{kj}^t, y_{kj}^t) \right. \\ &\qquad \left. - \eta \lambda_1 \sum_{i=1}^{t-1} (\boldsymbol{\theta}_{r-1,k}^t - \boldsymbol{\theta}^{i,*}) \boldsymbol{F}^i \right). \end{aligned} \tag{7}$$

With $\boldsymbol{\theta}_{r-1,k}^t$ for edge device k equal to $\boldsymbol{\theta}_{r-1}^t$ broadcast by the server, (7) is the same with (4). That is, in period t, the distributed implementation of EWC method is comprised of the following procedures: 1) The server sends $\boldsymbol{F}^i$ and an initial model $\boldsymbol{\theta}_0^t$ to all edge devices; 2) Edge device k updates local model following (6) and sends the update to the server; 3) The server aggregates the local model updates following (7) and sends the aggregated model to all edge devices; 4) Repeat the second and third procedures until achieving the optimal model $\boldsymbol{\theta}^{t,*}$.

During the distributed implementation described above, two key issues arise:

- **How to obtain T-FDM $\boldsymbol{F}^t$ after obtaining the optimal model $\boldsymbol{\theta}^{t,*}$?**
- **How to address the negative impact of spatial data heterogeneity among edge devices on achieving the optimal model $\boldsymbol{\theta}^{t,*}$?**

In the distributed implementation, the T-FDM has to be calculated in a distributed manner. To this end, edge device k computes a local T-FDM as

$$\boldsymbol{F}_k^t = \boldsymbol{E} \odot \frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}^{t,*}} \right) \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}^{t,*}} \right)^T, \tag{8}$$

and then sends it to the server for global aggregation:

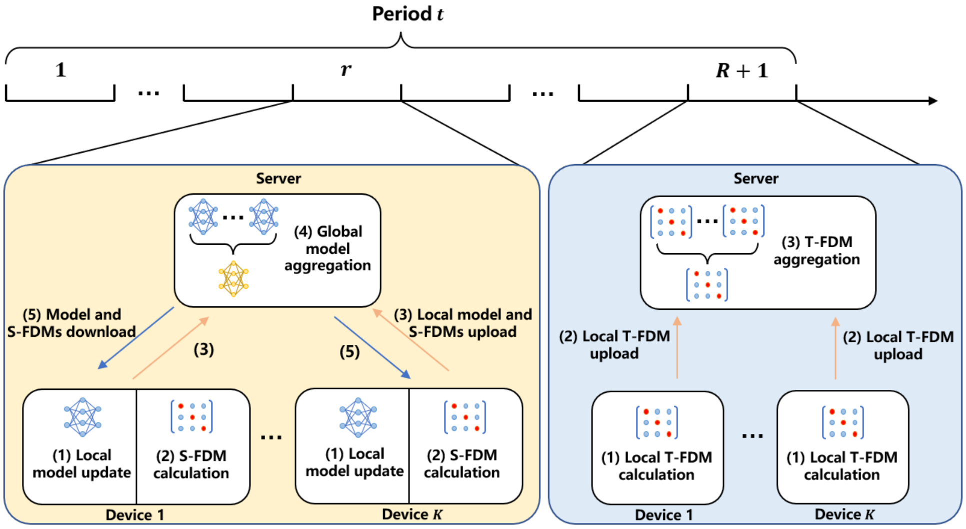


Figure 2. Federated continual learning flow with the proposed ST-EWC.

$$\begin{aligned}\boldsymbol{F}^t &= \sum_{k=1}^{K} \frac{n_{k,t}}{n_t} \boldsymbol{F}_k^t \\ &= \boldsymbol{E} \odot \frac{1}{n_t} \sum_{k=1}^{K} \sum_{j=1}^{n_{k,t}} \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}^{t,*}} \right) \left(\frac{\partial \ell(\boldsymbol{\theta}^{t,*}, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}^{t,*}} \right)^T , \end{aligned} \tag{9}$$

to finally attain a same T-FDM as (3).

To mitigate the adverse effects of spatial data heterogeneity, we further incorporate the EWC method in the spatial dimension. Specifically, we add an additional spatial regularization term for edge device k during its local model update in the r-th gradient descent, as follows:

$$\begin{aligned}\boldsymbol{\theta}_{r,k}^t =& \boldsymbol{\theta}_{r-1,k}^t - \eta \Bigg(\frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \triangledown \ell(\boldsymbol{\theta}_{r-1,k}^t, \boldsymbol{x}_{kj}^t, y_{kj}^t) \\ & + \lambda_1 \sum_{i=1}^{t-1} (\boldsymbol{\theta}_{r-1,k}^t - \boldsymbol{\theta}^{i,*}) \boldsymbol{F}^i \\ & + \lambda_2 \sum_{\substack{k'=1 \\ k' \neq k}}^{K} (\boldsymbol{\theta}_{r-1,k}^t - \boldsymbol{\theta}_{r-1,k'}^t) \boldsymbol{F}_{r-1,k'}^t \Bigg), \end{aligned} \tag{10}$$

where $\lambda_2 > 0$ is a weight factor for the spatial regularization term and $\boldsymbol{F}_{r-1,k'}^t$ represents the S-FDM, initialized as $\boldsymbol{F}_{0,k'}^t = \boldsymbol{0}$. For edge device k, its S-FDM is calculated as:

$$\begin{aligned}&\boldsymbol{F}_{r,k}^t = \\ &\boldsymbol{E} \odot \frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \left(\frac{\partial \ell(\boldsymbol{\theta}_{r,k}^t, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}_{r,k}^t} \right) \left(\frac{\partial \ell(\boldsymbol{\theta}_{r,k}^t, \boldsymbol{x}_{kj}^t, y_{kj}^t)}{\partial \boldsymbol{\theta}_{r,k}^t} \right)^T . \end{aligned} \tag{11}$$

Note that, the S-FDMs preserve the important knowledge learned from other devices. Hence, the local model update in (10) can prevent the updated model on device k from drifting away from this crucial knowledge, thereby promoting the convergence of local model toward a global optimum.

4.2. Learning Flow

Based on the design philosophy, we then elaborate on the learning flow with the proposed ST-EWC method. For clarity, we assume that after R communication rounds, the global model updated with (7) converges to the optimal model $\boldsymbol{\theta}^{t,*}$ in period t. As shown in Figure 2, the first R rounds are used for local model update following (10) and global model aggregation following (7), while the $R+1$-th round is used to update T-FDM F^t following (9). To clarify the design philosophy of our method, we only consider **one-time** local model update in the last subsection. Inspired by the idea in Federated Learning (McMahan et al., 2017) that increased local computation is beneficial for learning acceleration, we propose a more general learning flow in Algorithm 1, in which each local model is updated E times in the first R rounds.

4.2.1. Distributed Learning

In detail, we start with an initialization process in period t, during which the server broadcasts the S-FDM $\boldsymbol{F}^{t-1}$ and initial global model $\boldsymbol{\theta}_0^t$ to all edge devices. Then, the learning process lasts R rounds. In the r-th round, the following steps are executed.

(1) Local model update: Edge device k performs E local model updates, with the e-th update given by:

$$\begin{aligned}\boldsymbol{\theta}_{r,k}^{t,e} =& \boldsymbol{\theta}_{r,k}^{t,e-1} - \eta \Bigg(\frac{1}{n_{k,t}} \sum_{j=1}^{n_{k,t}} \nabla \ell(\boldsymbol{\theta}_{r,k}^{t,e-1}, \boldsymbol{x}_{kj}^{t}, y_{kj}^{t}) \\ &+ \lambda_1 \sum_{i=1}^{t-1} (\boldsymbol{\theta}_{r,k}^{t,e-1} - \boldsymbol{\theta}^{i,*}) \boldsymbol{F}^{i} \\ &+ \lambda_2 \sum_{\substack{k'=1 \\ k' \neq k}}^{K} (\boldsymbol{\theta}_{r,k}^{t,e-1} - \boldsymbol{\theta}_{r-1,k'}^{t}) \boldsymbol{F}_{r-1,k'}^{t} \Bigg), \end{aligned} \tag{12}$$

where the initial local model is set as $\boldsymbol{\theta}_{r,k}^{t,0} = \boldsymbol{\theta}_{r-1}^{t}$ and the final updated local model is further labeled as $\boldsymbol{\theta}_{r,k}^{t} = \boldsymbol{\theta}_{r,k}^{t,E}$. In fact, (12) is an extension of (10), generalizing from a one-time local model update to multiple updates. For broader applicability, the gradient descent in (12) can be replaced with more advanced optimizers such as stochastic gradient descent (SGD) or adaptive moment estimation (Adam).

(2) S-FDM calculation: After the local model update, edge device k calculates the S-FDM $\boldsymbol{F}_{r,k}^{t}$ according to (11).

(3) Local model and S-FDMs upload: Edge device k sends its updated local model $\boldsymbol{\theta}_{r,k}^{t}$ and S-FDM $\boldsymbol{F}_{r,k}^{t}$ to the server.

(4) Global model aggregation: After receiving all updated local models from K edge devices, the server follows (7) to aggregate these local models for a new global model $\boldsymbol{\theta}_{r}^{t}$.

(5) Model and S-FDMs download: The server sends the global model $\boldsymbol{\theta}_{r}^{t}$, along with the updated local models $\boldsymbol{\theta}_{r,k'}^{t}$ and S-FDMs $\boldsymbol{F}_{r,k'}^{t}$ from other devices, to edge device k.

4.2.2. Distributed T-FDM Calculation

After R rounds of distributed learning, the attained model θ_{R+1}^{t} converges to the optimal model $\boldsymbol{\theta}^{t,*}$. This optimal model is sent by the server to edge devices at the end of round R. Consequently, in round $R+1$, edge devices immediately compute the T-FDM in a federated manner:

(1) Local T-FDM calculation: After receiving $\boldsymbol{\theta}^{t,*}$, edge device k calculates its local T-FDM F_{k}^{t} according to (8).

(2) Local T-FDM upload: Edge device k uploads its local T-FDM F_{k}^{t} to the server.

(3) T-FDM aggregation: After receiving all the local T-FDMs from K devices, the server follows (9) to aggregate them for the distributed learning in the next period.

Algorithm 1 The proposed learning flow in the t-th period

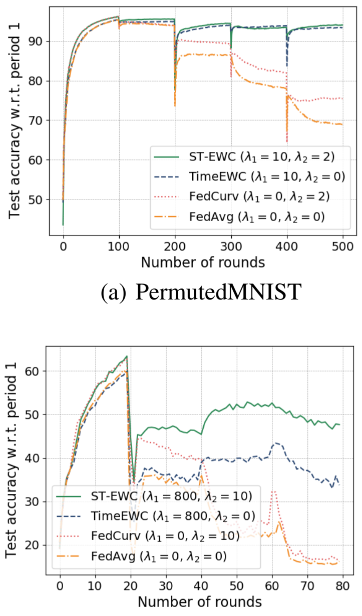
```
Distributed Learning:
Server sends $\boldsymbol{\theta}_0^t$ and $\boldsymbol{F}^{t-1}$ to devices;
for Round $r = 1$ to $R$ do
    for Device $k = 1$ to $K$ do
        for Local update $e = 1$ to $E$ do
            Following (12), device $k$ updates $\boldsymbol{\theta}_{r,k}^{t,e-1}$;
        end for
        Following (11), device $k$ computes $\boldsymbol{F}_{r,k}^{t}$;
        Device $k$ sends $\boldsymbol{\theta}_{r,k}^{t} = \boldsymbol{\theta}_{r,k}^{t,E}$ and $\boldsymbol{F}_{r,k}^{t}$ to server;
    end for
    Following (7), server aggregates $\{\boldsymbol{\theta}_{r,1}^{t}, ..., \boldsymbol{\theta}_{r,K}^{t}\}$ for $\boldsymbol{\theta}_{r}^{t}$;
    Server sends $\boldsymbol{\theta}_{r}^{t}$, $\boldsymbol{\theta}_{r,k'}^{t}$ and $\boldsymbol{F}_{r,k'}^{t}$ to device $k$;
end for
Distributed T-FDM Calculation:
for Device $k = 1$ to $K$ do
    Following (8), device $k$ computes $\boldsymbol{F}_{k}^{t}$;
    Device $k$ uploads $\boldsymbol{F}_{k}^{t}$ to server;
end for
Following (9), server aggregates $\{\boldsymbol{F}_{1}^{t}, ..., \boldsymbol{F}_{K}^{t}\}$ for $F^t$.
```

5. Experimental Setup and Results

5.1. Experimental Setup

In our experiments, we utilize the domain-incremental PermutedMNIST (Hashash et al., 2022) and PACS (Li et al., 2017) datasets to evaluate the performance of the proposed method. These datasets consist of a series of domains in which the input distribution changes over time, while the output label space remains unchanged. The PermutedMNIST dataset is a variant of the standard MNIST dataset, where the pixels of the handwritten digit images are randomly permuted in each period. This results in visually distinct versions of the same digits across periods, effectively simulating different data distributions in temporal dimension. For PermutedMNIST, we set the number of periods as $T = 5$ and number of devices is set to $K = 20$. To simulate a non-IID data distribution in spatial dimension, each device is assigned data from only two of the ten digit classes. A multi-layer perceptron (MLP) is used for training on PermutedMNIST, consisting of an input layer, two hidden layers with ReLU activations and Dropout regularization, and a final output layer. The MLP model is trained using the cross-entropy loss and SGD optimizer with a learning rate of $\eta = 0.1$. The number of rounds is set to $R = 100$ in each period and the number of local updates is set to $E = 1$ in each round.

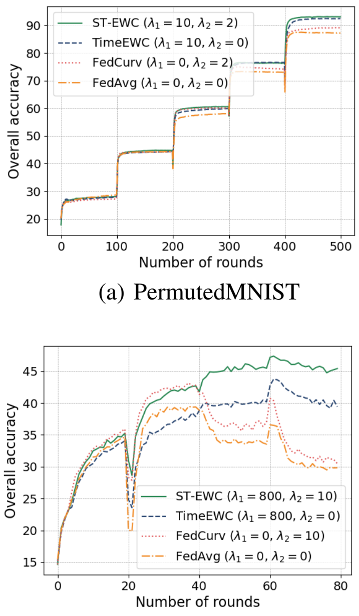
The PACS dataset consists of four distinct visual domains: Sketch, Cartoon, Art Painting, and Photo, and each domain shares the same set of object categories. In our setup, each domain corresponds to a distinct period, thereby resulting in $T = 4$ periods. We consider $K = 10$ devices, each of which is assigned data from three different object categories to simulate a non-IID data distribution in the spatial dimension. For PACS, we employ a convolutional neural network (CNN), comprised of two convolutional layers with ReLu activations, two max-pooling layers, and two fully connected



(a) PermutedMNIST

(b) PACS

Figure 3. Test accuracy w.r.t. period 1 vs. number of rounds.



(a) PermutedMNIST

(b) PACS

Figure 4. Overall accuracy vs. number of rounds.

layers with ReLU activations. The CNN model is trained using the cross-entropy loss and Adam optimizer with a learning rate of $\eta = 0.0001$. In each period, the number of rounds is set to $R = 20$, with each round comprising $E = 5$ local updates. To evaluate the effectiveness of our ST-EWC method, we compare it against the following baselines:

- FedAvg: FedAvg is a widely used FL algorithm (McMahan et al., 2017), which corresponds to our method with $\lambda_1 = 0$ and $\lambda_2 = 0$.
- FedCurv: FedCurv applies EWC only in the spatial dimension (Shoham et al., 2019), which corresponds to our method with $\lambda_1 = 0$.
- TimeEWC: TimeEWC is a distributed implementation of the proposed method in (Hashash et al., 2022), which applies EWC only in the temporal dimension and corresponds to our method with $\lambda_2 = 0$.

We further define three metrics to evaluate the test performance: test accuracy with respect to (w.r.t.) period 1, average accuracy across multiple periods, and overall accuracy on all test datasets. In period t, denote the current test dataset by $\mathcal{C}_t$ with data size c_t and accumulated test dataset by $\mathcal{O}_t \triangleq \{\mathcal{C}_1, ..., \mathcal{C}_t\}$ with data size $o_t = \sum_{i=1}^{t} c_i$. As a basis, we denote $\mathrm{Acc}((\cdot), (\cdot))$ as the test accuracy on dataset $(\cdot)$ evaluated using the model $(\cdot)$. Accordingly, in period t, the test accuracy w.r.t. period 1 is defined as

$$\mathrm{Acc}_t^1 = \mathrm{Acc}(\boldsymbol{\theta}_r^t, \mathcal{C}_1). \tag{13}$$

This metric is used to evaluate the forgetting degree of the r-th model $\boldsymbol{\theta}_r^t$ w.r.t. period 1. The overall accuracy of the r-th model $\boldsymbol{\theta}_r^t$ in period t is given by:

$$\widetilde{\mathrm{Acc}_t} = \mathrm{Acc}(\boldsymbol{\theta}_r^t, \mathcal{O}_T), \tag{14}$$

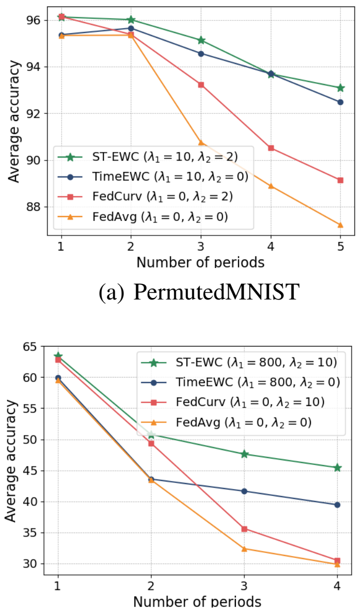
which evaluates the test performance of model $\boldsymbol{\theta}_r^t$ on the combined test datasets from all T periods. The average accuracy over the first period t is calculated as

$$\overline{\mathrm{Acc}_t} = \sum_{i=1}^{t} \frac{c_i}{o_t} \mathrm{Acc}(\boldsymbol{\theta}^{t,*}, \mathcal{C}_i), \tag{15}$$

which is a weighted average of the test accuracies of the converged model $\boldsymbol{\theta}^{t,*}$.

5.2. Result Analysis

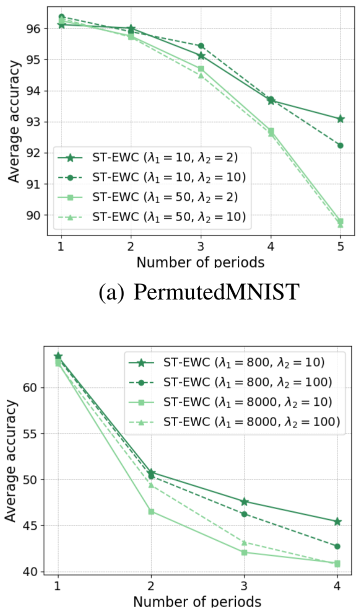
Figure 3 illustrates the test accuracy w.r.t. the first period. This figure shows 5-period MLP training for the PermutedMNIST dataset and 4-period CNN training for the PACS dataset, with each period consisting of 100 and 20 rounds, respectively. In the first period, the temporal regularization term in our ST-EWC is inactive, resulting in comparable learning performance to FedCurv. Similarly, TimeEWC behaves like FedAvg. That is, thanks to the effectiveness of spatial regularization term, both ST-EWC and FedCurv outperform TimeEWC and FedAvg in terms of convergence rate and test accuracy. In the subsequent periods, ST-EWC—enhanced by the temporal regularization term—surpasses FedCurv, while TimeEWC outperforms



(a) PermutedMNIST

(b) PACS

Figure 5. Average accuracy vs. number of periods.



(a) PermutedMNIST

(b) PACS

Figure 6. Average accuracy of ST-EWC.

FedAvg, exhibiting reduced forgetting with respect to the first period. Comparing ST-EWC and TimeEWC, we further observe that ST-EWC demonstrates superior knowledge retention, as it effectively addresses spatial data heterogeneous and better approaches the global optimal model in each period. Consequently, our ST-EWC achieves the best learning performance with respect to the first period and exhibits the lowest degree of forgetting of historical knowledge.

Figure 4 compares the overall accuracy of different methods, measured on the combined test datasets from all periods. For all methods, the overall accuracy increases as the number of periods grows. This is because, in the initial period, the model has only learned partial knowledge and thus performs poorly on future data. By contrast, in the final period, the model has accumulated and integrated knowledge from all periods, thereby significantly improving its overall accuracy. Notably, our ST-EWC consistently achieves the highest overall accuracy on both the PermutedMNIST and PACS datasets, with the performance gains becoming more pronounced as the number of periods increases. This highlights the effectiveness of ST-EWC in mitigating catastrophic forgetting and preserving learned knowledge over time. Additionally, Figure 5 illustrates the average accuracy of the four methods. As the number of periods increases, all methods suffer from forgetting, resulting in decreasing average accuracy. Nevertheless, our ST-EWC consistently outperforms the other baselines, achieving the hightest average accuracy.

In our ST-EWC method, the spatial regularization term focuses on learning from the current period's data, while the temporal regularization term emphasizes retaining knowledge from previous periods. As shown in Figure 6, achieving optimal average accuracy requires a balance between these two regularization terms. Specifically, when the temporal weight factor λ_1 is too large, the model becomes overly focused on preserving past knowledge, thereby hindering effective learning on current data and resulting in decreased average accuracy. Conversely, an excessively large spatial weight factor λ_2 overly emphasizes current performance, potentially interfering with temporal regularization and also degrading average accuracy. Ultimately, when both λ_1 and λ_2 are set to relatively small values, our method achieves a well-balanced trade-off between current learning and historical retention, yielding near-optimal average accuracy.

6. Conclusion

This paper has addressed the challenges of spatial and temporal heterogeneity in edge federated continual learning. To this end, we have proposed ST-EWC, a method that incorporates elastic weight consolidation constraints across both spatial and temporal dimensions. Experimental results have verified that ST-EWC effectively mitigates catastrophic forgetting, accelerates model convergence, and improves average accuracy, by striking a good balance between retaining historical knowledge and adapting to new data. These findings underscore the potential of our method for enabling adaptive and privacy-preserving edge intelligence.

Acknowledgements

This research is supported in part by the National Natural Science Foundation of China under Grant 62301222, 62271513, 62201024, 62201504, and in part by the National Key R&D Program of China under Grant 2024YFE0200700.

References

Aich, A. Elastic weight consolidation (EWC): Nuts and bolts. arXiv preprint arXiv:2105.04093, 2021.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383, 2(6):2, 2019.

Guo, K., Chen, Z., Yang, H. H., and Quek, T. Q. S. Dynamic scheduling for heterogeneous federated learning in private 5G edge networks. IEEE Journal of Selected Topics in Signal Processing, 16(1):26–40, 2022. doi: 10.1109/JSTSP.2021.3126174.

Hashash, O., Chaccour, C., and Saad, W. Edge continual learning for dynamic digital twins over wireless networks. arXiv preprint arXiv:2204.04795, 2022.

Jin, H., Zhang, P., Dong, H., Wei, X., Zhu, Y., and Gu, T. Mobility-aware and privacy-protecting QoS optimization in mobile edge networks. IEEE Transactions on Mobile Computing, 23(2):1169–1185, 2022.

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

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In proceedings of the IEEE international conference on computer vision, pp. 5542–5550, 2017.

Mallya, A. and Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 7765–7773, 2018.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In proceedings of the Artificial intelligence and statistics, pp. 1273–1282. PMLR, 2017.

Mei, Y., Yuan, L., Han, D.-J., Chan, K. S., Brinton, C. G., and Lan, T. Using diffusion models as generative replay in continual federated learning–what will happen? arXiv preprint arXiv:2411.06618, 2024.

Qi, D., Zhao, H., and Li, S. Better generative replay for continual federated learning. arXiv preprint arXiv:2302.13001, 2023.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. iCaRL: Incremental classifier and representation learning. In proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 2001–2010, 2017.

Shoham, N., Avidor, T., Keren, A., Israel, N., Benditkis, D., Mor-Yosef, L., and Zeitak, I. Overcoming forgetting in federated learning on non-IID data. arXiv preprint arXiv:1910.07796, 2019.

Usmanova, A., Portet, F., Lalanda, P., and Vega, G. Federated continual learning through distillation in pervasive computing. In proceedings of the IEEE International Conference on Smart Computing, pp. 86–91. IEEE, 2022.

Wang, Z., Wu, F., Yu, F., Zhou, Y., Hu, J., and Min, G. Federated continual learning for edge-AI: A comprehensive survey. arXiv preprint arXiv:2411.13740, 2024.

Wei, Y., Wang, X., Guo, K., Yang, H. H., and Chen, X. Fedds: Data selection for streaming federated learning with limited storage. In proceedings of the IEEE Wireless Communications and Networking Conference, pp. 1–6, 2024. doi: 10.1109/WCNC57260.2024.10570766.

Wu, F., Qu, Y., Wu, T., Dong, C., Guo, K., Wu, Q., and Guo, S. Participant and sample selection for efficient online federated learning in UAV swarms. IEEE Internet of Things Journal, 2024.

Yoon, J., Jeong, W., Lee, G., Yang, E., and Hwang, S. J. Federated continual learning with weighted inter-client transfer. In proceedings of the International Conference on Machine Learning, pp. 12073–12086. PMLR, 2021.

You, C., Guo, K., Yang, H. H., and Quek, T. Q. Hierarchical personalized federated learning over massive mobile edge computing networks. IEEE Transactions on Wireless Communications, 22(11):8141–8157, 2023.