Edge Continual Learning for Dynamic Digital Twins over Wireless Networks

Omar Hashash, Christina Chaccour, and Walid Saad
Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, USA.
Email:{omarnh, christinac, walids}@vt.edu

***Abstract*—Digital twins (DTs) constitute a critical link between the real-world and the metaverse. To guarantee a robust connection between these two worlds, DTs should maintain *accurate* representations of the physical applications, while preserving *synchronization* between real and digital entities. In this paper, a novel edge continual learning framework is proposed to accurately model the evolving affinity between a physical twin (PT) and its corresponding cyber twin (CT) while maintaining their utmost synchronization. In particular, a CT is simulated as a deep neural network (DNN) at the wireless network edge to model an autonomous vehicle traversing an episodically dynamic environment. As the vehicular PT updates its driving policy in each episode, the CT is required to concurrently adapt its DNN model to the PT, which gives rise to a de-synchronization gap. Considering the *history-aware* nature of DTs, the model update process is posed a dual objective optimization problem whose goal is to jointly minimize the loss function over all encountered episodes and the corresponding de-synchronization time. As the de-synchronization time continues to increase over sequential episodes, an elastic weight consolidation (EWC) technique that regularizes the DT history is proposed to limit de-synchronization time. Furthermore, to address the *plasticity-stability* tradeoff accompanying the progressive growth of the EWC regularization terms, a modified EWC method that considers fair execution between the historical episodes of the DTs is adopted. Ultimately, the proposed framework achieves a simultaneously accurate and synchronous CT model that is robust to *catastrophic forgetting*. Simulation results show that the proposed solution can achieve an accuracy of 90% while guaranteeing a minimal de-synchronization time.**

***Index Terms*—dynamic digital twins, continual learning, elastic weight consolidation (EWC), synchronization, massive twinning**

I. Introduction

Digital twins (DTs) are central to the digital transformation that we are currently witnessing. In essence, DTs will be a driving force for an end-to-end digitization beyond the Industry 4.0 borders [1]. By extending the fundamental DT concept towards massive twinning, a plethora of complex real-world DTs are expected to emerge [2]. This is crucial for enabling anticipated Internet of Everything (IoE) applications such as extended reality and connected robotics and autonomous systems [3]. Guaranteeing a high-fidelity representation of such DTs that have a dynamic nature constrains the underlying communication network with a set of stringent wireless requirements. For instance, extreme reliability, hyper-mobile connectivity, near-zero latency, and high data rates are necessary to sustain high quality-of-service (QoS) for such dynamic DTs. One measure that can bring us closer to guaranteeing such requirements is the integration of DTs with an intelligent edge backbone. Furthermore, an artificial intelligence (AI)-native edge plays a pivotal role in supporting fully-autonomous IoE services in 6G networks [4]. In fact, deploying DTs for autonomous IoE applications operating in non-stationary real-world scenarios necessitates adopting an intelligent wireless edge that can seamlessly adapt the digital duplicate to the changes in the underlying application states.

Several works [5]–[8] studied the implementation of DTs at the edge in an attempt to meet their stringent requirements and meet their challenges. The authors in [5] proposed coupling deep reinforcement learning (DRL) with transfer learning (TL) to solve the DT-edge placement and migration problems in dynamic environments. The work in [6] proposed a dynamic congestion control scheme to manage the stochastic demand in DT-edge networks. The authors in [7] developed a solution for the dynamic association problem in DT-edge networks while balancing the accuracy and cost of a blockchain-federated learning (FL) scheme. In [8], the authors studied the use of a DT for capturing the air-ground network dynamics that assist network elements in designing incentives for a DT-edge FL scheme. While the works in [5]–[8] shed light on interesting issues that accompany DT-edge networks from different dynamic perspectives, they consider major limiting assumptions. The works in [6]–[8] assume the DTs to operate under ideal stationary conditions. Nonetheless, realistic DTs that emulate a digital replica of novel IoE applications operate in dynamic non-stationary environments. Meanwhile, even though the work in [5] considers non-stationary conditions when using TL, it disregards the knowledge gained with the *continuous* evolution of the DTs. Clearly, there is a lack in works that comprehensively consider the practical operation of evolving DTs under real dynamic considerations.

In fact, enabling a dynamic DT in a massive twinning context is governed by defining the duality between physical and digital entities. This duality is triggered on multiple fronts that require merging new dimensions and metrics, including: (i) *synchronization* between the physical twin (PT) and the cyber twin (CT) to preserve the real-time interaction and diminish the possibility of interrupting the CT operations and simulations, (ii) *accuracy* of the CT in reflecting the real-time PT status by maintaining a precise duplicate throughout the various phases experienced by the PT, and (iii) *history-awareness* of the evolving DT states by incorporating the knowledge attained from past experiences into future states. *Hence, a fundamental step towards achieving a massive twin-*

This research was supported by the U.S. National Science Foundation under Grants CNS-2007635 and CNS-1836802.

ning framework can be accomplished by enabling simultaneously synchronous, accurate, and history-aware DTs that can continuously operate in dynamic non-stationary environments.

The main contribution of this paper is a novel edge continual learning (CL) approach to accurately model the evolving affinity between a PT and its corresponding CT in a real dynamic environment, while preserving their synchronization. In particular, a CT is simulated as a deep neural network (DNN) at the wireless network edge to model an autonomous vehicle traversing through an episodically dynamic environment. This vehicular PT is equipped with Internet of Things (IoT) sensors that capture the relevant data used in determining the autonomous driving actions. As the distribution of the IoT data changes with each episode, the vehicular agent adapts its driving policy accordingly. Consequently, the CT must update its model to preserve the DT accuracy, which leads to a de-synchronization gap between the twins. While taking the history-awareness of DTs into account, the CT model update process is formulated as a dual objective optimization problem to minimize the loss over all encountered episodes while reducing de-synchronization time. To limit the progressive increase in de-synchronization time over episodes, we propose a CL-based elastic weight consolidation (EWC) technique that contracts the ongoing historical experience as regularization terms into the learning cost. As the impact of EWC regularization on CT evolution increases with number of episodes encountered, we use a variant of EWC to address the *plasticity-stability* effect in DTs. *To the best of our knowledge, this is the first work to model DTs in a dynamic environment by addressing the synergy between synchronization, accuracy, and ongoing history.* Simulation results show that our proposed method to model DTs can outperform state-of-art benchmarks by providing near optimal accuracy measures over all encountered episodes within the minimal de-synchronization time.

II. System Model

A. DT-Edge Model

Consider a DT system in which the PT is an autonomous vehicle connected to a base station (BS), as shown in Fig. 1. To empower this vehicle with autonomous functionality, it is equipped with a large set of IoT sensors that continuously capture data used in executing real-time driving decisions. For instance, the type of data generated by the IoT sensors can range from intra-vehicular analytics to inter-vehicular range sensing that are necessary to fine-tune the vehicle's speed. Moreover, the BS is equipped with a mobile edge computing (MEC) server that runs a real-time DT simulation of the autonomous driving vehicle. In particular, the vehicular CT simultaneously replicates the PT's driving decisions as a function of the IoT data uploaded to the DT-edge layer. This real-time DT simulation is managed in a deep learning framework, where a DNN model $\mathcal{M}_0$ that represents the CT is trained over a dataset $\mathcal{D}_0$ comprising historical actions exhibited by the PT. The upload time of IoT data to the DT-edge layer is assumed to be negligible.

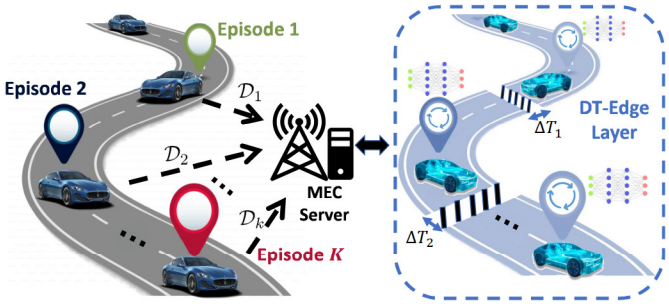


Fig. 1: Illustration of the edge dynamic digital twins system performing a simulation of the physical system in a non-stationary environment

As a result of the dynamic changes in the space, time, as well as the driving conditions; the environment experienced by the vehicle will vary in a *non-stationary* fashion. These dynamic and non-stationary variations are reflected in the generated IoT data readings and their respective probability distributions. Thus, the PT and its respective CT are required to adjust their driving policy to cope with those variations.

Henceforth, we consider a PT to be operating in a dynamic environment, where the environment changes over a set $\mathcal{K}$ of K sequential time episodes [9]. For tractability, although the inter-episode variations are non-stationary, we assume that the intra-episode environment variations are stationary. In our model, the learning agent has full knowledge of the time frames, i.e., the start and end of each episode. As a result of dynamic changes in the PT environment, the learning agent of the PT autonomously updates its driving policy. *Consequently, considering the history-aware nature of DTs, the CT should adapt its DNN model to encompass the current episode $k \in \mathcal{K}$ while reflecting the knowledge from its past experiences.*

At the start of each episode k, a dataset $\mathcal{D}_k = \{\boldsymbol{x}_{ki}, y_{ki}\}_{i=1}^{i=I_k}$ of the vehicle actions is collected, where $I_k = |\mathcal{D}_k|$ and the pair $\{\boldsymbol{x}_{ki}, y_{ki}\} \sim \mathcal{P}_k$ represents the i^{th} data vector of the IoT sensor measurements and the corresponding decision taken by the vehicle, during episode k, respectively. Then, this dataset $\mathcal{D}_k$ is uploaded to the DT-edge layer to update a DNN model $\mathcal{M}_k$ in episode k.

B. Learning over the DT-Edge

The model $\mathcal{M}_k$ is assumed to have l fully-connected layers where $l = 1, 2, \ldots, L$. In addition, each layer is composed of a_{lm} neurons, where $m = m_1, m_2, \ldots, m_l$ is the number of neurons in each layer. Moreover, $\mathcal{M}_k$ is characterized with weights $\boldsymbol{w}_k^{n_k} \in \mathbb{R}^n$ reached upon performing n_k training iterations, with $n = \sum_{l=1}^{L-1} m_l m_{l+1}$ being the dimension size of the model weights vector. Due to the aforementioned dynamic spatio-temporal conditions, the data probability distribution $\mathcal{P}_k$, where $\mathcal{D}_k \sim \mathcal{P}_k$, experiences non-stationary variations with respect to k episodes. Hence, to procure a comprehensive update that describes the entire PT evolution, the CT should adapt its model to the current episode k while reflecting the experience gained over the preceding $k-1$ episodes. In accordance, we define a loss function that measures the

performance of $\mathcal{M}_k$ over all collected PT actions, as follows:

$$\mathcal{L}(\boldsymbol{w}_k^{n_k}, \mathcal{A}_k) = \frac{1}{|\mathcal{A}_k|} \sum_{j=0}^{k} \sum_{i=1}^{I_k} \psi(\boldsymbol{w}_k^{n_k}, \boldsymbol{x}_{ji}, y_{ji}), \quad (1)$$

where $\mathcal{A}_k = \bigcup_{j=0}^{k} \mathcal{D}_j$ is the *non-iid* accumulated dataset of all PT actions until current episode k and $\psi(\cdot)$ is a generic loss function. The training process over the *non-iid* dataset is then carried out using gradient descent (GD) iterations to minimize the loss function as follows: $\boldsymbol{w}_k^{n_k} \leftarrow \boldsymbol{w}_k^{n_k-1} - \eta \nabla \mathcal{L}(\boldsymbol{w}_k^{n_k-1}, \mathcal{A}_k)$, where η is the learning rate. Since the DT simulation is an explicitly synchronous process, the CT cannot perform simulation and learning tasks simultaneously. Thus, we assume that the simulation process is halted throughout the updating process of $\mathcal{M}_k$, which gives rise to a synchronization gap between the twins. To capture the time during which the DT is out-of-service, we define the *de-synchronization time* as the time needed to update $\mathcal{M}_k$ during each episode k, as follows:

$$\Delta T_k(n_k) = \frac{|\mathcal{A}_k|}{f} \Lambda n_k, \quad (2)$$

where Λ is the number of CPU cycles required to train one data sample, and f is the CPU frequency of the MEC server. Next, we formulate a dual objective optimization problem to guarantee *episodic history-aware CT model updates to accurately represent the PT, while limiting the de-synchronization time between the twins.*

C. Problem Formulation

Our goal is to jointly maximize the DT's accuracy and minimize the corresponding de-synchronization time over each episode. In particular, we tend to guarantee a synchronous DT simulation where the CT model accurately adapts to each episode sequentially while abiding by the knowledge acquired from previous experience. Thus, it is necessary to find the optimal number of iterations need to learn the corresponding weights while achieving an optimal tradeoff between enhancing model accuracy and limiting de-synchronization time. Thus, at the beginning of each non-stationary episode k the learning agent needs to optimize the following problem, in a sequential fashion:

$$\min_{n_k} \quad \alpha \mathcal{L}(\boldsymbol{w}_k^{n_k}, \mathcal{A}_k) + (1-\alpha)\Delta T_k(n_k), \quad (3a)$$

$$\text{s.t.} \quad \alpha \in [0,1], n_k \in \mathbb{Z}, \quad (3b)$$

where α is a bias coefficient that controls the accuracy and synchronization tradeoff in our application.

This is a dual objective optimization problem that minimizes the loss function and the de-synchronization time to maintain an accurate CT model in real-time, by controlling the number of GD iterations in each episode. Solving problem (3) using classical machine learning schemes is challenging due to the growth of the accumulated dataset size that is included in the GD computations over each episode – a unique feature of DTs over wireless networks.

As the size of the accumulated dataset increases with the number of episodes encountered, the de-synchronization time grows linearly with the additional time required to update the model. Given that the episodes have equal importance and contribution to the learning model, the complete ongoing history of experiences should be considered in the loss function. On the one hand, disregarding the previous DT experiences violates the history-awareness requirement of our DT. On the other hand, training over the accumulated dataset will jeopardize synchronization and violate DT latency requirements. Thus, it is necessary to propose a novel solution that can ensure a swift model adaptation in each episode while fulfilling the synchronization and history-aware DT requirements. Next, we develop a novel edge CL paradigm that balances the accuracy and de-synchronization time in the CT update process while taking the historical experience into consideration.

III. Edge CL for Dynamic DTs

In this section, we present a CL-based solution that can mitigate the progressive increase in de-synchronization time which accompanies the growth of accumulated dataset size. In general, CL is recognized as an incremental learning scheme that enables agents to learn individual tasks sequentially [10], without the need to learn over all of the previously encountered tasks' data. Accordingly, by considering the DT performance over each episode as an individual task, CL can promote individual episode learning while preserving the knowledge gained from encountered episodes. Thus, by alleviating the need to learn over the accumulated dataset, the increase in de-synchronization time can diminish noticeably.

With respect to our DTs system, our solution will embrace the EWC technique that was introduced to mitigate the *catastrophic forgetting* phenomenon that captures the affinity of DNNs to forget the weights related to previous tasks upon consecutive model updates [11]. In particular, EWC enables selective regularization of model weights that are essential to reflect the CT experience in each episode. Hence, a major portion of the DT history is conveyed within the DNN weights and contracted as quadratic penalty terms inserted into the loss function. Thus, regularizing the history can limit *catastrophic forgetting* and the increase in de-synchronization time.

First, the DNN weights are acquired by employing a sequential Bayesian approach. In particular, to estimate the posterior distribution of the DNN weights given the accumulated data $\mathcal{A}_k$ and, assuming that the episodes are independent, the log posterior distribution is defined according to the Bayes rule [11] as: $\log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_k) \propto \log p(\mathcal{D}_k|\boldsymbol{w}_k^{n_k}) + \log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_{k-1})$, where $\log p(\mathcal{D}_k|\boldsymbol{w}_k^{n_k})$ is the loss in the current episode k and $\log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_{k-1})$ is the prior probability. Thus, training the model $\mathcal{M}_k$ is carried out by a minimization over the negative log posterior function to yield the solution weights $\boldsymbol{w}_k^*$ in episode k as:

$$\boldsymbol{w}_k^* = \arg\min_{\boldsymbol{w}_k^{n_k}} \left(-\log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_k)\right), \quad (4)$$

Here, (4) represents the DT simulation that reflects the DT's state in episode k. Furthermore, the ongoing DT history is implicitly reflected throughout the prior term that provides a bias of the experience gained over the preceding $k-1$ episodes. In essence, the term $\log p(\mathcal{D}_k|\boldsymbol{w}_k^{n_k})$ is related to the CT simulation error while minimizing the

loss function with respect to $\mathcal{D}_k$ and $\boldsymbol{w}_k^{n_k}$. The prior term $\log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_{k-1})$ is intractable as shown in [12], which limits the evaluation of the posterior distribution as a direct term. To evaluate this term, one can formulate its Laplace approximation [13], and, thus, approximate it through its second order Taylor expansion in the neighborhood of $\boldsymbol{w}_{k-1}^*$ as: $\log p(\boldsymbol{w}_k^{n_k}|\mathcal{A}_{k-1}) \approx \frac{1}{2}(\boldsymbol{w}_k^{n_k} - \boldsymbol{w}_{k-1}^*)^\top \boldsymbol{H}(\boldsymbol{w}_{k-1}^*)(\boldsymbol{w}_k^{n_k} - \boldsymbol{w}_{k-1}^*)$, where $\boldsymbol{H}(\boldsymbol{w}_{k-1}^*) = \frac{\partial^2}{\partial \boldsymbol{w}_k^2} \log\, p(\boldsymbol{w}_k|\mathcal{A}_{k-1})|_{\boldsymbol{w}_k = \boldsymbol{w}_{k-1}^*}$ is the Hessian of the log prior probability function evaluated at $\boldsymbol{w}_{k-1}^*$. Moreover, the first and second terms of the Taylor expansion are neglected since they are evaluated near the optimal solution $\boldsymbol{w}_{k-1}^*$. Furthermore, the Hessian in this case can be represented in terms of the Fisher information matrix that shows the importance of each weight in this episode. Without loss of generality, the Fisher information matrix is defined as $\boldsymbol{F}_k = -\mathbb{E}_{(\boldsymbol{x}_k, y_k)\sim\mathcal{D}_k}(\boldsymbol{H}(\boldsymbol{w}_k^*))$. Assuming that the weight parameters are independent of each other, we can estimate the Fisher information matrix in terms of its diagonal elements while setting the rest to zero [14]. Hence, after quantifying the terms related to the posterior probability, we define the *recursive loss function* that combines both current and EWC losses in episode k as follows:

$$\mathcal{L}(\boldsymbol{w}_k^{n_k}) = \mathcal{L}(\boldsymbol{w}_k^{n_k}, \mathcal{D}_k) + \sum_{j=0}^{k-1} \sum_{d} \frac{\lambda}{2} F_{j,dd} \left(w_{k,d}^{n_k} - w_{j,d}^*\right)^2, \tag{5}$$

where $\mathcal{L}(\boldsymbol{w}_k^{n_k}, \mathcal{D}_k)$ is the CT simulation error on $\mathcal{D}_k$, $F_{j,dd}$ is the diagonal value of the Fisher information matrix in episode j that relates to the weight d, λ is a hyperparameter that determines the relative weighting of learning the new episode in comparison to remembering the previous episodes experience, $w_{k,d}$ is the weight in episode k, and $w_{j,d}^*$ is the optimal solution weight for parameter d in episode j.

The EWC loss term in (5) can substantially increase with the number of episodes encountered, which consequently jeopardizes the model's robustness. Hence, the model's ability to *continuously* update its weights will be dominated by the rigid effect of the regularization terms. Eventually, this will lead to a *stable* CT model that is constrained by its history and cannot evolve further. On the other hand, when the effect of regularization is minimal, the model is susceptible to losing its experience which leads to a *plastic* CT model that is vulnerable to *catastrophic forgetting*. This phenomenon is known as the *stability-plasticity dilemma* [15]. To address this dilemma, we must equip the CT model with higher degrees of freedom to smoothly address stability and plasticity effects along different episodes. To achieve that, we propose adopting a modified EWC version called EWC++ [16]. Here, the Fisher information matrix is estimated through a moving average after calculating the Fisher matrix in each episode k. Accordingly, the averaged Fisher elements are estimated at the end of each episode k as: $\tilde{F}_{k,dd} = \gamma F_{k,dd} + (1-\gamma)\tilde{F}_{(k-1),dd}$, where γ is a hyperparamater. After setting the value of γ and calculating the averaged Fisher elements at each episode, the modified recursive loss function can be expressed as:

$$\tilde{\mathcal{L}}(\boldsymbol{w}_k^{n_k}) = \mathcal{L}(\boldsymbol{w}_k^{n_k}, \mathcal{D}_k) + \sum_{d} \frac{\lambda}{2} \tilde{F}_{(k-1),dd} \left(w_{k,d}^{n_k} - w_{(k-1),d}^*\right)^2.$$

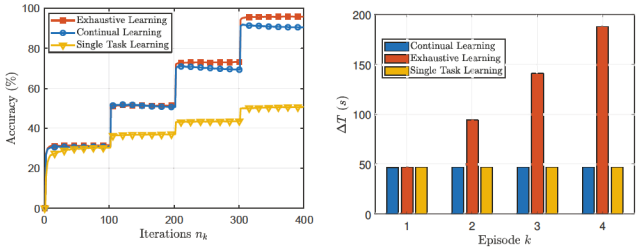
This loss deals with the current and previous episode Fisher values along with the previous weights $\boldsymbol{w}_{k-1}^*$ only. Essentially, implementing this modified loss function can yield a representative CT model that *continually* learns while countering the stability-plasticity effects and mitigating the increase in de-synchronization time. Hence, this modified loss function will replace that of (3a) to solve problem (3). Henceforth, the problem in (3) can be then interpreted as a modified loss minimization problem by controlling n_k that achieves the desired balance between accuracy and de-synchronization of the DTs according to α. Solving this modified loss minimization problem through GD iterations will yield the optimal n_k^* and its corresponding $\boldsymbol{w}_k^*$ for episode k.

IV. Simulation Results and Analysis

For our simulations, we consider a CT operating over a MEC server at a BS for $k = 4$ sequential episodes. The CT must *sequentially* learn a classification task of identifying the handwritten digits from 0 to 9 of the permuted MNIST dataset, which is a variant of the standard MNIST dataset having image pixels permuted independently for each episode. For our model, we use a multi layer perceptron (MLP) having $L = 2$ hidden layers, each having $m_l = 256$ neuron units with ReLu activations $\forall l = \{1, 2\}$. We set $\psi(\cdot)$ as the cross-entropy function. Unless otherwise stated, we set $I_k = 15000$, $\eta = 0.01$, $\Lambda = 125440$ cycles/sample, $f = 4\,\text{GHz}$, $\lambda = 75000$, and $\gamma = 0.5$ [16], [17]. A dataset of 1000 images was procured, by collecting images from the different permutations used to test the trained model. In our experiments, we compare the proposed CL solution presented with two benchmark methods: a) An *exhaustive learning* scheme that accumulates all the training data throughout its history and current episode, b) A *single task learning* approach that solely relies on the training data seen in the current episode.

Fig. 2a shows the achieved accuracy on the test set over 400 training iterations for the different methods, where each 100 iterations refers to one sequential episode. From Fig. 2a, we observe that the proposed CL solution achieves near optimal performance reaching an accuracy of 90%, compared to the optimal solution of 95% resulting from the exhaustive learning scheme. Meanwhile, the single task learning method did not retain all of its knowledge between episodes which resulted in an accuracy of only 50%. The results showcase how the proposed CL method is capable of learning incrementally without revisiting data from previous episodes.

Fig. 2b presents the de-synchronization time in each episode. Here, the exhaustive learning scheme achieves a high accuracy, at the expense of a significantly increased de-synchronization time. In this figure, we observe de-synchronization times of 45 s and 190 s in episodes 1 and 4, respectively. In contrast, the proposed CL solution has a clear advantage by maintaining its de-synchronization time at 45 s throughout the 4 episodes, while reaching a near optimal accuracy. Although single task learning has a similar de-synchronization time to CL, this comes at the expense of a deteriorated accuracy of 50% at the end of the 4 episodes.



(a) (b)

Fig. 2: a) Accuracy (%) versus the number of iterations n_k, b) De-synchronization time upon training over 4 sequential episodes.

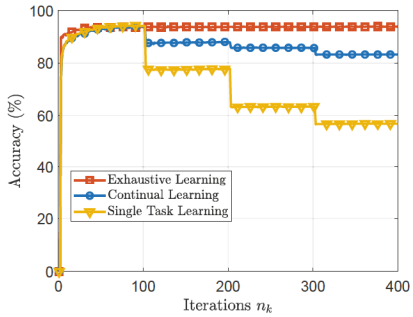


Fig. 3: Accuracy (%) over first episode versus iterations n_k

Fig. 3 compares the model robustness to *catastrophic forgetting* upon training the different methods over 4 sequential episodes. This is verified by measuring the model's accuracy on the first episode after training on each episode successively. Initially, it can be seen that all the methods achieve a similar accuracy of 94% over the first episode. As the number of episodes increases, we can see that the proposed CL is capable of reaching an accuracy of 84% in contrast to a 58% attained accuracy by single task learning. Clearly, unlike the exhaustive method, the CL methodology is a lightweight scheme that permits accumulating the gained knowledge without the need to revisit data from preceding episodes. That said, the proposed approach achieves an accuracy that is only 14% lower than that of exhaustive learning. Henceforth, this showcases the resilience of the proposed CL method to *catastrophic forgetting* despite its minimal de-synchronization time.

Fig. 4 presents the relationship between the training loss and de-synchronization time, where the CL solution proposed is implemented in a single episode for the different values of n_k. Fig. 4 demonstrates the presence of an explicit tradeoff between training loss and the de-synchronization time when solving for the optimal value of n_k^* that is dependent on α. By varying α for $0 \leq \alpha \leq 1$, ΔT can increase with the increase in n_k reaching 47 s for 100 iterations performed. Consequently, by increasing n_k, the parameters converge to their optimal values $\boldsymbol{w}_k^*$, which minimizes the training loss accordingly.

V. Conclusion

In this paper, we have proposed a novel edge CL framework to enable a robust synergy between a CT and PT in a dynamic environment. To guarantee an accurate and synchronous CT, we have posed its model update process as a dual objective optimization problem. We have adopted an EWC technique

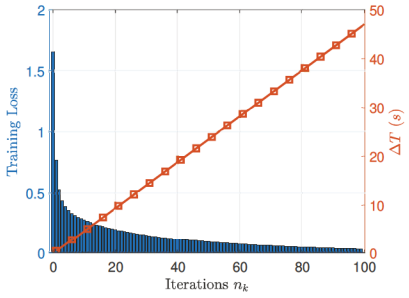


Fig. 4: Training loss and de-synchronization time ΔT (s) versus the number of iterations n_k during one episode.

to overcome the increase in the de-synchronziation time. To address the stability-plasticity tradeoff accompanying the progressive growth of the regularization terms, we have proposed a modified CL paradigm that considers a fair execution between historical episodes. Ultimately, we have shown that the proposed approach can achieve an accurate and synchronous CT that can balance between adaptation and stability.

References

[1] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, Jan. 2022.

[2] B. Han and H. D. Schotten, "Multi-sensory HMI to enable digital twins with human-in-loop: A 6G vision of future industry," *arXiv preprint arXiv:2111.10438*, 2021.

[3] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, P. Popovski, and M. Debbah, "Seven defining features of terahertz (THz) wireless systems: A fellowship of communication and sensing," *IEEE Communications Surveys & Tutorials*, 2022.

[4] C. Chaccour and W. Saad, "Edge Intelligence in 6G Systems," in *6G Mobile Wireless Networks*. Springer, 2021, pp. 233–249.

[5] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 219–16 230, Nov. 2021.

[6] X. Lin, J. Wu, J. Li, W. Yang, and M. Guizani, "Stochastic digital-twin service demand with edge response: An incentive-based congestion control approach," *IEEE Transactions on Mobile Computing*, 2021.

[7] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5098–5107, July 2020.

[8] W. Sun, N. Xu, L. Wang, H. Zhang, and Y. Zhang, "Dynamic digital twin and federated learning with incentives for air-ground networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 321–333, Jan./Feb. 2022.

[9] H. Sun, W. Pu, X. Fu, T.-H. Chang, and M. Hong, "Learning to continuously optimize wireless resource in a dynamic environment: A bilevel optimization perspective," *IEEE Transactions on Signal Processing*, 2022.

[10] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

[12] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, Aug. 2018.

[13] F. Huszár, "On quadratic penalties in elastic weight consolidation," *arXiv preprint arXiv:1712.03847*, 2017.

[14] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

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

[16] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 532–547.

[17] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the International Conference on Machine Learning (ICML)*, Aug. 2017, pp. 3987–3995.