Federated Class Incremental Learning: A Pseudo Feature Based Approach Without Exemplars

Min Kyoon Yoo[1] and Yu Rang Park[1]

Department of Biomedical Systems Informatics, Yonsei University College of Medicine, Seoul, Republic of Korea
minkyoon_da@yonsei.ac.kr, yurangpark@yuhs.ac

Abstract. Federated learning often assumes that data is fixed in advance, which is unrealistic in many real-world scenarios where new data continuously arrives, causing catastrophic forgetting. To address this challenge, we propose FCLPF (Federated Class Incremental Learning with Pseudo Features), a method that uses pseudo features generated from prototypes to mitigate catastrophic forgetting. Our approach reduces communication costs and improves efficiency by eliminating the need for past data and avoiding computationally heavy models like GANs. Experimental results on CIFAR-100 show that FCLPF achieves an average accuracy of 51.87% and an average forgetting of 9.62%, significantly outperforming existing baselines with an average accuracy of 47.72% and forgetting of 20.46%. On TinyImageNet, FCLPF achieves 37.56% accuracy and 3.14% forgetting, compared to the baselines' 27.69% accuracy and 24.46% forgetting, demonstrating the superior performance of FCLPF.

Keywords: Federated class incremental learning · Continual learning · Incremental learning · Catastrophic forgetting · Federated learning

1 Introduction

Federated learning [9] is decentralized machine learning that does not require data centralization . The absence of centralized data collection allows for the preservation of privacy. It has recently attracted considerable attention in both research and industry. It is employed in fields such as healthcare [18] and the development of local devices [8].

In conventional federated learning, it is assumed that data remains fixed over time. This assumption is not valid in the real world. For example, in healthcare, A disease classification model may require incorporating new disease data over time. Therefore, the model must be capable of learning new data without forgetting previously learned information.

Catastrophic forgetting [14] occurs when the addition of new data causes the model to forget previously learned information due to class imbalance. In the centralized setting, this issue is studied under incremental learning [23]. Several methods have been proposed to mitigate catastrophic forgetting [2, 7, 21, 28].

Federated class incremental learning (FCIL) [4] addresses the aforementioned challenges by preventing catastrophic forgetting when learning new data in a federated learning environment. A significant proportion of the existing proposed methods make use of rehearsal memory [3,4,17]. Nevertheless, past data retention can prove challenging for devices with limited storage [20], potentially raising concerns about data privacy.

To address these issues, we propose FCLPF (Federated Class Incremental Learning with Pseudo Features), a novel approach that generates pseudo features without relying on rehearsal memory. The proposed FCLPF generates pseudo features by utilizing only prototypes and data in the client. To create a pseudo feature, only prototype types need to be exchanged between the client and server, reducing communication costs and preserving privacy. This method prevents catastrophic forgetting and is efficient in federated learning environments. The code is available on GitHub[1].

2 Related Work

Federated Class Incremental Learning. The objectives of Federated Class Incremental Learning (FCIL) is to preserve previously learned knowledge while incorporating new data into the global model.

The studies GLFC [4] and LGA [3] employ a proxy server to mitigate catastrophic forgetting. However, having a proxy server raises possible privacy issues and increases communication costs. Furthermore, contrary to common federated learning assumptions, both studies allow clients to share data. Moreover, rehearsal memory is necessary for both methods, which is impractical for devices with little storage. Additionally, FedRCIL [17] employs self-supervised learning (SSL) and knowledged distillation to FCIL, yet it also leverages rehearsal memory.

In certain instances, past data is not accessible, and the retention of such data is challenging for devices with limited memory. Several methods of using GAN [5] to replace past data have been proposed. FedCIL [19] generates synthetic data for global model consolidation by using locally trained generative models. MFCL [1] and TARGET [27] also demonstrate the feasibility of FCIL without the use of rehearsal memory. However, MFCL and TARGET employ a centralized server to train GAN, which incurs a significant computational cost and a high communication cost for sending and receiving models.

Exemplar-Free Class Incremental Learning Class incremental learning (CIL) algorithms can be employed effectively when new data is introduced. Catastrophic forgetting occurs when learning new data and is a significant challenge in machine learning. This is due to the lack of past data, a crucial factor in the learning process. Several CIL methods have recently been developed to combat this problem [2, 7, 21, 28]. Numerous exemplar-based approaches have

[1] `https://github.com/DigitalHealthcareLab/24FCLPF.git`

been proposed. However, these methods present challenges in low-memory environments and may raise privacy concerns.

To address these challenges, Exemplar-Free CIL (EFCIL) has been proposed as a potential solution [24, 26, 30]. These methods do not retain past data. A significant number of EFCIL methods employ regularization techniques to facilitate incremental updates to the deep model, and distillation is used to preserve past knowledge [13].

Some techniques, such as PASS [30], employ prototype augmentation to prevent catastrophic forgetting across incremental states; other techniques, such as ABD [24], use picture inversion to produce pseudo-samples of previous classes. It then uses linear head fine-tuning and importance-weighted feature distillation to distinguish between task features from the past and present without the requirement for rehearsal memory.

For IL2A [29], the features are generated using the information from the class distribution to create the features of previous classes. In FeTrIL [16], the feature extractor is fixed, and pseudo-features are generated straightforwardly. Our method fixes the feature extractor like FeTrIL and generates pseudo features. By fixing the feature extractor, only the fully connected (FC) layer needs to be exchanged, which significantly reduces the communication cost, making this an efficient and suitable method in federated class incremental learning. To the best of our knowledge, our study is the first to apply pseudo features to FCIL.

3 Method

We propose FCLPF, a federated class incremental learning that uses pseudo features to replace rehearsal memory. Previous studies are mostly with rehearsal memory, and those without rehearsal memory create and use GANs, which leads to very high computational and communication costs. Therefore, we created pseudo features to replace rehearsal memory. This study is the first to apply pseudo features to FCIL. We propose a new way to create pseudo features using Principal Component Analysis (PCA) [25]. Our method is illustrated in Figure 1.

Initially, the client trains the feature extractor and the FC layer. At the end of the initial task, the server aggregates the class-specific prototypes from all clients, performing a sample-weighted prototype averaging based on the number of samples per class for each client. This ensures that the class distribution of all clients is reflected in the global model.

Following the initial task, the feature extractor is frozen. The server sends only the FC layers and global prototypes to the clients. The clients use the received prototypes to generate pseudo features, replacing rehearsal memory. This approach reduces the parameters that need to be shared with the central server by sharing only FC layers and prototypes, significantly reducing communication costs in a federated learning environment.

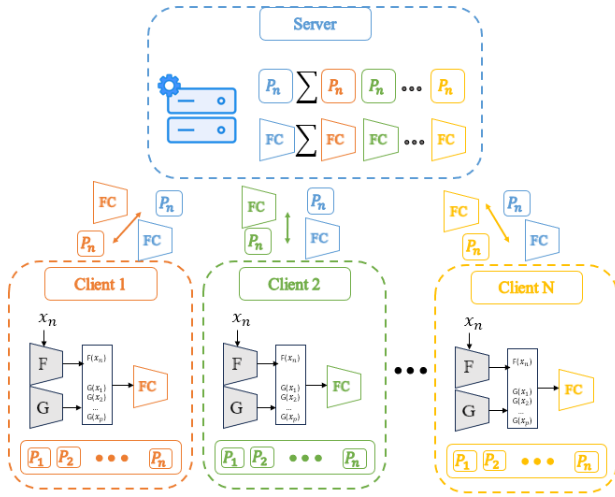


Fig. 1: Overview of our method. (P_n) is the prototype of the nth class, and FC denotes the fully connected (FC) layer. $F(x_n)$ represents the feature of x_n, whereas $G(x_p)$ denotes the pseudo feature for the past class. Only prototypes and FC layers are exchanged between the client and server. The server aggregates the prototype and FC layer.

Prototype. We don't save the past data. We create a pseudo feature. To create the pseudo feature, we used the prototype. A prototype was created using the following formula:

$$\boldsymbol{\mu}_c = \frac{1}{|X_c|} \sum_{\boldsymbol{x} \in X_c} f_c(\boldsymbol{x}) \tag{1}$$

,where $f_c(\boldsymbol{x})$ denotes feature vector of sample x belonging to class c. $\boldsymbol{\mu}_c$ denotes prototype for class c. We extracted the features for each class and averaged them. We sent the pseudo features created by each client in the first round of the task to the center and averaged them to create a single prototype. The prototype created by the center was sent to each client so that they could create pseudo features within the client.

Closest prototype. We compute the new class that is most similar to the old class we want to create. The closest prototype was calculated using the formula as follows:

$$C_{\text{closest}} = \arg \min_{c_{\text{new}} \in C_{\text{new}}} \text{dist}(\boldsymbol{\mu}_{c_{\text{old}}}, \boldsymbol{\mu}_{c_{\text{new}}}) \tag{2}$$

, where C_{new} represents the set of new classes, and $\boldsymbol{\mu}_{c_{\text{old}}}$ and $\boldsymbol{\mu}_{c_{\text{new}}}$ represents the prototype of the old class and the prototype of the new class, respectively. $\text{dist}(\cdot, \cdot)$ denotes the distance metric. In this study, we employed the cosine similarity as the distance metric. We then calculated the difference between the old prototype $\boldsymbol{\mu}_{c_{\text{old}}}$ and the closest new class prototype $\boldsymbol{\mu}_{c_{\text{closest}}}$:

$$\Delta_{c_{\text{old}},c_{\text{closest}}} = \boldsymbol{\mu}_{c_{\text{old}}} - \boldsymbol{\mu}_{c_{\text{closest}}}. \tag{3}$$

PCA Projection. We applied PCA to the prototype of the old class $\boldsymbol{\mu}_{c_{\text{old}}}$, the prototypes of all new classes $\boldsymbol{\mu}_{C_{\text{new}}}$ to find the principal axes. We projected $\Delta_{c_{\text{old}},c_{\text{closest}}}$ onto these principal axes and restored the original dimensions as follows:

$$\text{PCA}_k(\{\boldsymbol{\mu}_{c_{\text{old}}}\} \cup \{\boldsymbol{\mu}_{C_{\text{new}}}\}) = P_k \tag{4}$$

$$\Delta^{\text{projected}}_{c_{\text{old}},c_{\text{closest}}} = \left(\Delta_{c_{\text{old}},c_{\text{closest}}} \times P_k^T\right) \times P_k \tag{5}$$

, where P_k represents the matrix of the k principal components that explains the most variance.

Pseudo feature generation. We generated an old class pseudo feature. We added the $\Delta^{\text{projected}}_{c_{\text{old}},c_{\text{closest}}}$ to $f_{\text{closest}}(\boldsymbol{x})$. We created a pseudo feature as follows:

$$f_{c_{\text{old}}}(\boldsymbol{x}) = f_{C_{\text{closest}}}(\boldsymbol{x}) + \Delta^{\text{projected}}_{c_{\text{old}},c_{\text{closest}}} \tag{6}$$

, where $f_{c_{\text{old}}}(\boldsymbol{x})$ represents the pseudo feature of the old class generated by using $f_{\text{closest}}(\boldsymbol{x})$ which is the feature vector of sample $\boldsymbol{x}$ belonging to class C_{closest}.

4 Experiment

Setting. We experimented with our method CIFAR-100 [10], TinyImageNet [11], and SuperImageNet [1] to demonstrate its effectiveness. We used Resnet18 [6] as the backbone model and compared our model with Fedavg [15], FedProx [12], and MFCL [1]. The parameter settings are identical to those described in MFCL [1]. For CIFAR-100, the number of clients is 50, and the local epoch is 10. For TinyImageNet and SuperImageNet, the number of clients is 100, and the local epoch is 10. The SGD optimizer was utilized. The data for each task was distributed among the clients using Latent Dirichlet Allocation (LDA) [22] with $\alpha = 1$. The learning rate for all tasks starts at 0.1 and decreases exponentially to 0.01 over time.

Incremental setting. We configured the experiment the same as the previous Exemplar-free class incremental learning setting [29–31]. The CIFAR-100 dataset was tested in three scenarios:

Scenario 1: 50 initial classes and 5 incremental learning states of 10 classes each
Scenario 2: 50 initial classes and 10 incremental learning states of 5 classes each
Scenario 3: 40 initial classes and 20 incremental learning states of 3 classes each

The TinyImageNet dataset was also tested in three scenarios:

Scenario 1: 100 initial classes and 5 incremental learning states of 20 classes each

Scenario 2: 100 initial classes and 10 incremental learning states of 10 classes each
Scenario 3: 100 initial classes and 20 incremental learning states of 5 classes each

The SuperImageNet dataset was additionally tested in two separate scenarios:

Scenario 1: 30 initial classes and 5 incremental learning states of 4 classes each
Scenario 2: 30 initial classes and 10 incremental learning states of 2 classes each

Metric. We compared the performance metric with average accuracy. Average accuracy averages the accuracy of all tasks. It is calculated as follows:

$$\widetilde{A} = \frac{1}{T} \sum_{t=1}^{T} A_t \tag{7}$$

where A_t is the accuracy of the task t, T is the total number of tasks, and $\widetilde{A}$ is the average accuracy.

Average forgetting evaluates how much the model forgets the previous task while learning a new task. Forgetting (f_t) of task t is defined as the difference between the highest performance on task t and the performance after all training. The average forgetting is calculated by averaging all the f_t for tasks 1 to $T-1$ Using the following formula:

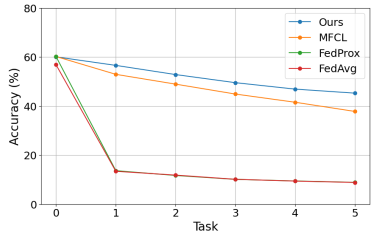
$$\widetilde{f} = \frac{1}{T-1} \sum_{t=1}^{T-1} f_t \tag{8}$$

4.1 Results

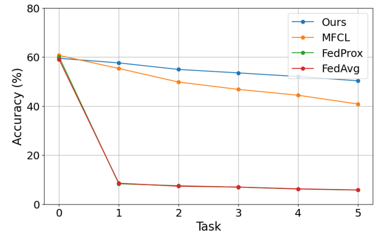
The proposed method demonstrated superior performance compared to the baseline both datasets across all scenarios. Figure 2 illustrates the performance for each task. The method demonstrated the highest performance for all tasks. The detailed experimental results are presented in Table 1, 2, and 3 , which compare the average accuracy and average forgetting across different scenarios. Our method outperformed the other methods in all scenarios.

In the CIFAR-100 dataset, the average accuracy of FCLPF is 51.87%, 51.06%, and 49.30% for scenarios 1, 2, and 3, respectively. In contrast, the other federated class incremental learning method, MFCL, achieved 47.72%, 43.17%, and 37.13%, respectively. The performance gap between our method and the other methods widened as the number of tasks increased. Our method achieved the best average forgetting results. FCLPF's average forgetting was significantly lower, with values of 3.14%, 4.00%, and 4.10% in scenarios 1, 2, and 3, respectively. This indicates better retention of previous knowledge compared to MFCL, which had average forgetting values of 20.46%, 33.57%, and 43.73% in the same scenarios.

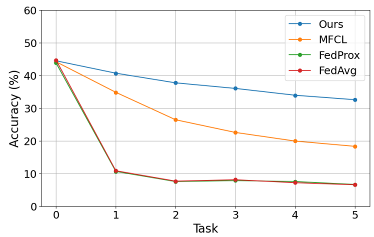
In Tiny ImageNet, our proposed FCLPF demonstrated superior performance compared to other baselines. Furthermore, the performance difference between



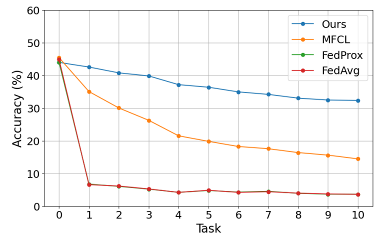
(a) CIFAR Scenario 1



(b) CIFAR Scenario 2



(c) TinyImageNet Scenario 1



(d) TinyImageNet Scenario 2

Fig. 2: Comparison of model performance across different tasks and datasets

Table 1: Comparison of Average Accuracy and Average Forgetting for CIFAR-100

Scenario	Method	Average Accuracy (%)	Average Forgetting (%)
S1	**Ours**	**51.87 ± 5.23**	**9.62 ± 2.47**
	MFCL	47.72 ± 7.40	20.46 ± 29.63
	FedAvg	18.42 ± 17.25	65.97 ± 32.74
	FedProx	19.00 ± 18.47	66.12 ± 32.79
S2	**Ours**	**51.06 ± 4.65**	**10.35 ± 4.52**
	MFCL	43.17 ± 8.78	33.57 ± 28.02
	FedAvg	10.79 ± 15.29	79.74 ± 26.49
	FedProx	10.90 ± 15.59	80.04 ± 26.52
S3	**Ours**	**49.30 ± 5.40**	**10.88 ± 6.95**
	MFCL	37.13 ± 9.57	43.73 ± 22.18
	FedAvg	6.73 ± 12.02	85.99 ± 20.24
	FedProx	6.74 ± 12.06	85.71 ± 20.49

our method and other baselines increased with the increased number of tasks. In Tiny ImageNet, the average accuracy of our method was 37.56%, 37.05%, and 37.01% in scenarios 1, 2, and 3, respectively. In contrast, MFCL achieved 27.69%, 23.67%, and 16.55%, respectively. The average forgetting is also dominant over MFCL, and as the task size increases, our method's average forgetting becomes more pronounced compared to other baselines.

In SuperImageNet, our proposed FCLPF showed superior performance compared to other baselines. In Scenario 1, our method achieved an average accuracy of 65.37%, while MFCL only achieved 51.21%. Similarly, in Scenario 2, our method recorded 65.10%, significantly outperforming MFCL's 47.44%. Moreover, our method consistently exhibited lower average forgetting compared to other baselines. In Scenario 1, our method had an average forgetting of only 8.36%, while MFCL's forgetting reached 28.41%. Similarly, in Scenario 2, our

Table 2: Comparison of Average Accuracy and Average Forgetting for TinyImageNet

Scenario	Method	Average Accuracy (%)	Average Forgetting (%)
S1	**Ours**	**37.56 ± 4.04**	**3.14 ± 2.89**
	MFCL	27.69 ± 9.12	24.46 ± 21.33
	FedAvg	14.16 ± 13.71	49.67 ± 24.99
	FedProx	14.00 ± 13.39	49.47 ± 25.01
S2	**Ours**	**37.05 ± 3.95**	**4.00 ± 3.47**
	MFCL	23.67 ± 9.24	38.06 ± 20.61
	FedAvg	8.36 ± 11.58	63.30 ± 21.39
	FedProx	8.28 ± 11.32	63.24 ± 21.41
S3	**Ours**	**37.01 ± 3.68**	**4.10 ± 4.17**
	MFCL	16.55 ± 8.00	48.18 ± 18.88
	FedAvg	4.67 ± 9.02	73.64 ± 17.69
	FedProx	4.61 ± 8.85	73.46 ± 17.85

Table 3: Comparison of Average Accuracy and Average Forgetting for SuperImageNet

Scenario	Method	Average Accuracy (%)	Average Forgetting (%)
S1	**Ours**	**65.37 ± 5.37**	**8.36 ± 3.33**
	MFCL	51.21 ± 11.04	28.41 ± 33.99
	FedAvg	18.99 ± 23.87	67.36 ± 34.93
	FedProx	19.04 ± 23.74	68.43 ± 35.42
S2	**Ours**	**65.10 ± 5.43**	**11.67 ± 5.32**
	MFCL	47.44 ± 10.92	40.87 ± 33.34
	FedAvg	8.36 ± 11.58	63.30 ± 21.39
	FedProx	10.63 ± 19.44	78.64 ± 29.38

method showed an average of 11.67% forgetting, while MFCL's forgetting was much higher at 40.87%. These results confirm that FCLPF is more effective in mitigating forgetting and maintaining higher accuracy in SuperImageNet scenarios than other baselines.

Figure 3 compares the typical number of parameters exchanged per round for each task by method. As the initial task parameters are similar, we compare the shared parameters after the initial task. In contrast to other methods, our approach does not exchange feature extractors, only FC layer. Furthermore, since we employ pseudo features, we are not required to exchange models, such as GANs, which is a significant advantage. Consequently, our method is more efficient in terms of communication costs.

The experimental results demonstrate that FCLPF achieves high accuracy and is an effective method to prevent catastrophic forgetting. Furthermore, the minimal number of shared parameters makes it a highly efficient algorithm.

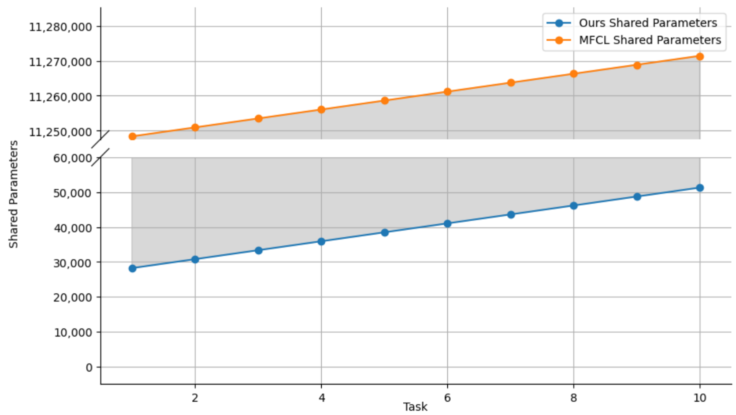


Fig. 3: Comparison of shared parameters across tasks for different methods. The figure illustrates the number of shared parameters per round for each task.

5 Conclusion

In this study, we proposed FCLPF (Federated Class Incremental Learning With Pseudo Features), a novel method that employs pseudo features in federated class incremental learning for privacy-preserving and communication efficiency without relying on rehearsal memory. Our approach leverages PCA to generate pseudo features. Our experimental results on CIFAR-100 and TinyImageNet showed that our model is superior in all tasks compared to other baseline models. The key innovation of FCLPF is creating pseudo features to replace rehearsal memory, resulting in low communication costs since only prototypes need to be exchanged. The results presented in this study demonstrate that the proposed method prevents catastrophic forgetting and facilitates efficient federated learning.

For future work, we plan to explore training the feature extractor in conjunction with pseudo features to enhance the model's performance. Furthermore, even though we presently average the prototypes, our goal is to explore different approaches for creating and applying prototypes to improve the accuracy and robustness of the pseudo features.

Overall, FCLPF represents a significant advancement in federated class incremental learning. The study addresses the limitations of rehearsal memory and presents an efficient method for reducing communication costs while using pseudo-features.

Acknowledgement. This research was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (No. P0023675, HRD Program for Industrial Innovation) and a grant of the Korea Machine Learning Ledger Orchestration for Drug Discovery Project (K-MELLODDY), funded by the Ministry of Health & Welfare and Ministry of Science and ICT, Republic of Korea (Grant Number: RS-2024-12345678).

References

1. Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M., Avestimehr, S.: A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. Advances in Neural Information Processing Systems **36** (2024)
2. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European conference on computer vision (ECCV). pp. 233–248 (2018)
3. Dong, J., Li, H., Cong, Y., Sun, G., Zhang, Y., Van Gool, L.: No one left behind: Real-world federated class-incremental learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
4. Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., Zhu, Q.: Federated class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10164–10173 (2022)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 831–839 (2019)
8. Imteaj, A., Thakker, U., Wang, S., Li, J., Amini, M.H.: A survey on federated learning for resource-constrained iot devices. IEEE Internet of Things Journal **9**(1), 1–24 (2021)
9. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
11. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
12. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
13. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., Van De Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(5), 5513–5533 (2022)
14. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
16. Petit, G., Popescu, A., Schindler, H., Picard, D., Delezoide, B.: Fetril: Feature translation for exemplar-free class-incremental learning. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 3911–3920 (2023)

17. Psaltis, A., Chatzikonstantinou, C., Patrikakis, C.Z., Daras, P.: Fedrcil: Federated knowledge distillation for representation based contrastive incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3463–3472 (2023)
18. Qayyum, A., Ahmad, K., Ahsan, M.A., Al-Fuqaha, A., Qadir, J.: Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. IEEE Open Journal of the Computer Society **3**, 172–184 (2022)
19. Qi, D., Zhao, H., Li, S.: Better generative replay for continual federated learning. arXiv preprint arXiv:2302.13001 (2023)
20. Ravaglia, L., Rusci, M., Nadalini, D., Capotondi, A., Conti, F., Benini, L.: A tinyml platform for on-device continual learning with quantized latent replays. IEEE Journal on Emerging and Selected Topics in Circuits and Systems **11**(4), 789–802 (2021)
21. Rebuffi, S.A., Kolesnikov, A., Sperl, G., 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)
22. Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. arXiv preprint arXiv:2003.00295 (2020)
23. Schlimmer, J.C., Fisher, D.: A case study of incremental concept induction. In: Proceedings of the Fifth AAAI National Conference on Artificial Intelligence. pp. 496–501 (1986)
24. Smith, J., Hsu, Y.C., Balloch, J., Shen, Y., Jin, H., Kira, Z.: Always be dreaming: A new approach for data-free class-incremental learning-supplementary materials (appendix)-
25. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems **2**(1-3), 37–52 (1987)
26. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6982–6991 (2020)
27. Zhang, J., Chen, C., Zhuang, W., Lyu, L.: Target: Federated class-continual learning via exemplar-free distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4782–4793 (2023)
28. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13208–13217 (2020)
29. Zhu, F., Cheng, Z., Zhang, X.Y., Liu, C.l.: Class-incremental learning via dual augmentation. Advances in Neural Information Processing Systems **34**, 14306–14318 (2021)
30. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)
31. Zhu, K., Zhai, W., Cao, Y., Luo, J., Zha, Z.J.: Self-sustaining representation expansion for non-exemplar class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9296–9305 (2022)