Dynamic Participation in Federated Learning: Benchmarks and a Knowledge Pool Plugin

Ming-Lun Lee[1], **Fu-Shiang Yang**[1], **Cheng-Kuan Lin**[1], **Yan-Ann Chen**[2], **Chih-Yu Lin**[3], **Yu-Chee Tseng**[1]

[1]Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan
[3]Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan

Abstract

Federated learning (FL) enables clients to collaboratively train a shared model in a distributed manner, setting it apart from traditional deep learning paradigms. However, most existing FL research assumes consistent client participation, overlooking the practical scenario of dynamic participation (DPFL), where clients may intermittently join or leave during training. Moreover, no existing benchmarking framework systematically supports the study of DPFL-specific challenges. In this work, we present the first open-source framework explicitly designed for benchmarking FL models under dynamic client participation. Our framework provides configurable data distributions, participation patterns, and evaluation metrics tailored to DPFL scenarios. Using this platform, we benchmark four major categories of widely adopted FL models and uncover substantial performance degradation under dynamic participation. To address these challenges, we further propose Knowledge-Pool Federated Learning (KPFL), a generic plugin that maintains a shared knowledge pool across both active and idle clients. KPFL leverages dual-age and data-bias weighting, combined with generative knowledge distillation, to mitigate instability and prevent knowledge loss. Extensive experiments demonstrate the significant impact of dynamic participation on FL performance and the effectiveness of KPFL in improving model robustness and generalization.

1 Introduction

Federated learning (FL) (McMahan et al. 2017) enables collaborative training of a server model across multiple clients without sharing private data. However, most FL solutions assume that clients participate consistently throughout the training process. In mobile or unstable environments, clients may dynamically join, leave, and even rejoin training. Such dynamic participation (DP) is common among battery-powered, solar-operated edge devices, as well as participants in social or vehicular networks (Du et al. 2020).

To address this realistic yet underexplored setting, we investigate dynamic participation in federated learning (DPFL). DP introduces several key challenges: First, it significantly degrades training performance due to inconsistent data availability and highly variable client updates, which can slow convergence (Gu et al. 2021; Ruan et al. 2021), increase instability (Lan et al. 2024), and result in catastrophic forgetting (McCloskey and Cohen 1989; Goodfellow et al. 2013; Kirkpatrick et al. 2017; Yang et al. 2024). Second, participation patterns are inherently unpredictable, as some clients drop out temporarily or permanently, and some join only in late stages. Third, the non-IID nature of client data further exacerbates these issues (Li et al. 2022).

Dynamic client behaviors have been partially addressed in previous works. Flexible device participation is considered in (Ruan et al. 2021), which studies how inactive devices and mid-training arrivals or departures affect convergence and extends FedAvg (McMahan et al. 2017) with simple weighting. MIFA (Gu et al. 2021) mitigates client unavailability by caching gradients but applies static aggregation weights. Federated Continual Learning (FCL) (Usmanova et al. 2021; Ma et al. 2022; Yang et al. 2024) focuses on preserving and adapting knowledge across evolving tasks. FCL methods handle consistent feature space expansion when new

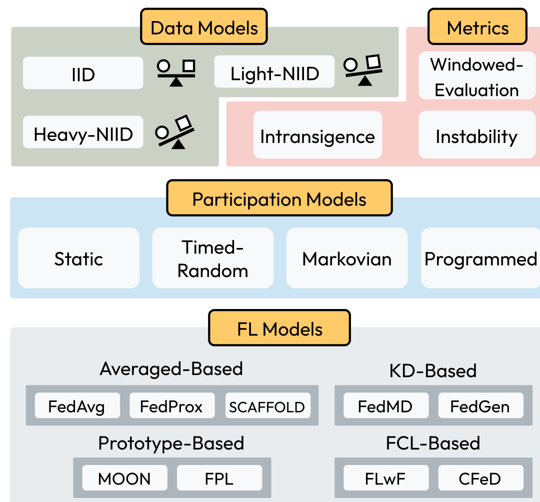

Figure 1: The proposed DPFL benchmarking framework.

*Accepted at the 1st Workshop on Federated Learning for Critical Applications (FLCA), in conjunction with AAAI26 tasks are introduced. In contrast, DPFL addresses the unstable feature space and prevents knowledge collapse caused by fragmented updates from intermittent clients. Learning dynamics have also been explored in FedAvg and Asynchronous Federated Learning (AFL) (Xu et al. 2023; Zhou et al. 2024; Zhang et al. 2023). FedAvg uses random client sampling to scale with large client pools, while AFL accommodates varying client availability to improve aggregation efficiency. In addition, recent effort (Düsing and Cimiano 2025) prototypes FL as a digital learning platform. However, these works do not explicitly model the participation patterns that characterize DPFL. Although several platforms exist, such as TFF (TensorFlow Federated Authors 2019), FedML (He et al. 2020), Flower (Beutel et al. 2020), and PySyft (Ziller et al. 2021), they lack support for DP scenarios. To our knowledge, there is currently no benchmark capable of evaluating the unique challenges posed by DPFL.

In this work, we fill this gap by introducing the first open-source DPFL platform that explicitly models client dynamics (Fig. 1). The platform provides (i) controllable data models (e.g., IID, light- and heavy-NIID), (ii) realistic and probabilistic DP models, and (iii) a set of DPFL-specific evaluation metrics. Our benchmark suite systematically evaluates the impact of DPFL across nine state-of-the-art FL models spanning four widely-used categories.

Building on insights from this benchmark, we propose knowledge-pool FL (KPFL), a generic plugin that maintains historical model states for both active and idle clients. By applying age-aware and data-bias–aware weighting with generative distillation, KPFL substantially mitigates instability and knowledge loss. We demonstrate successful integration of KPFL into all nine evaluated models, achieving substantial improvements in robustness under various DP scenarios.

2 Review: Benchmarking Targets

We benchmark 9 FL methods in 4 groups. (i) *average-based* methods: FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), and SCAFFOLD (Karimireddy et al. 2020), (ii) *knowledge distillation (KD)-based* methods: FedMD (Li and Wang 2019) and FedGen (Zhu, Hong, and Zhou 2021), (iii) *prototype-based* methods: MOON (Li, He, and Song 2021) and FPL (Huang et al. 2023), (iv) *federated continual learning (FCL)-based* methods: FLwF (Usmanova et al. 2021) and CFeD (Ma et al. 2022).

3 Design of the DPFL Framework

3.1 Modeling Data Heterogeneity

To introduce data heterogeneity, we modify the label distributions across clients and classes following the Dirichlet distribution (Li et al. 2022). For each class k, we sample $p_k \sim \mathrm{Dir}_N(\alpha)$, where N is the number of clients, p_k is a vector of length N, and α is the concentration parameter. Client j receives a proportion $p_{k,j}$ of the class-k data, $j = 1, 2, \ldots, N$. Based on α, we categorize data heterogeneity into three levels: (i) *IID*: $\alpha = 100$; the distributions are approximately uniform. (ii) *Light-NIID*: $\alpha = 1.0$; the distributions are moderately non-IID. (iii) *Heavy-NIID*: $\alpha = 0.1$; the distributions are highly imbalanced.

3.2 Modeling Participation Dynamics

Let $C = \{c_0, c_1, \ldots, c_{N-1}\}$ denote the complete set of clients, where c_i represents the i-th client, $i = 0, 1, \ldots, N-1$. Let $C_t \subseteq C$ be the subset of clients participating in training round t. Our platform supports several models.

Timed-Random. Each client c_i participates in round t with a time-varying probability $p_i(t)$, modeling geographically distributed, solar-powered clients.

Markovian. Each client alternates between active and inactive states based on a Markov transition matrix $\mathcal{P} = \left[\begin{smallmatrix} p_{0\to0} & p_{0\to1} \\ p_{1\to0} & p_{1\to1} \end{smallmatrix}\right]$, modeling energy-saving Discontinuous Reception (DRX) (Liang et al. 2020, 2018, 2016) in LTE/5G.

Programmed. Custom participation patterns can be imported, allowing replay of real-world client behaviors.

3.3 Performance Metrics

While current FL benchmarks primarily focus on final-round accuracy, we define several cross-round quantitative metrics to better understand the impact of DPFL.

Windowed Evaluation (↑): Let ψ_i denote a performance metric of the global model measured in the i-th round. The windowed evaluation metric, WE_t, is computed as a statistical property of ψ_i (e.g., mean, variance; mean by default) over a window of ω consecutive rounds.

Intransigence to DP (↓): This metric quantifies the average performance gap between dynamic and static participation, reflecting a model's robustness to client dynamics.

$$IDP = \frac{1}{T_f} \textstyle\sum_{i=1}^{T_f} (\psi_i^* - \psi_i) \tag{1}$$

Instability due to DP (↓): This metric assesses the stability of a given performance index ψ_i within a window $i \in [T_{s_1} : T_{s_2}]$ by calculating the average absolute deviation between the actual and its regressed values $\{\tilde{\psi}_i\}$.

$$ID_{T_{s_1},T_{s_2}} = \left(\textstyle\sum_{i=T_{s_1}+1}^{T_{s_2}} |\tilde{\psi}_i - \psi_i| \right) / (T_{s_2} - T_{s_1}) \tag{2}$$

4 Knowledge-Pool Federated Learning

KPFL is designed as a model-agnostic plug-in to mitigate DPFL challenges while remaining fully compatible with existing FL models. Fig. 2 shows the architecture of KPFL.

4.1 Knowledge Pool

The server maintains a knowledge pool $\mathcal{KP}$, which captures the evolving knowledge of both active and idle clients to support adaptive and robust global model training. For each client c_i, the server maintains the following information:

θ_i	the most recent local model update of c_i
τ_i	the timestamp of c_i's most recent state transition (active/idle)
$n_{i,j}$	the number of data items of class j held by c_i
z_i	the local auxiliary object associated with θ_i
aa_i, ia_i	active age and idle age indicators
aw_i, dw_i	age weight and data bias weight

We propose a *dual-age weighting* mechanism to address adaptiveness and forgetfulness. For each active client c_i, we

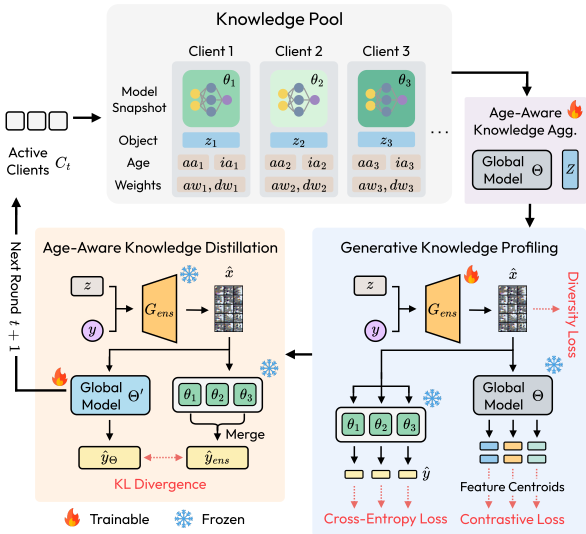

Figure 2: The Knowledge-Pool Federated Learning framework.

define its active age as $aa_i = t - \tau_i$, where t is the current time (or round). A smaller aa_i implies greater freshness. Conversely, for each idle client c_i, we define its idle age as $ia_i = t - \tau_i$, where a larger idle age indicates more severe staleness. The age weight of client c_i at time t is given by:

$$aw_i = \begin{cases} e^{\lambda_{aa} \cdot aa_i} & \text{if } c_i \text{ is active} \\ e^{\lambda_{ia} \cdot ia_i} & \text{if } c_i \text{ is idle} \end{cases} \tag{3}$$

where λ_{aa} and λ_{ia} are the age decay coefficients.

To further balance statistical heterogeneity, each client c_i is assigned a data bias weight $dw_i = \sum_{j=1}^{K} n_{i,j}/N_j$, where $N_j = \sum_{c_i} n_{i,j}$ is the total number of class j samples. This design allows clients with larger representative datasets to contribute proportionally more during aggregation.

4.2 Age-Aware Knowledge Aggregation

Via the age and data bias weights, we aggregate the first version of global model Θ and a global auxiliary object $\mathcal{Z}$.

$$(\Theta, \mathcal{Z}) = \textstyle\sum_{c_i} W_i \cdot (\theta_i, z_i) \tag{4}$$

$W_i = aw_i \cdot \varepsilon_i + dw_i$ represents an aggregated weight of c_i using a weighting factor $\varepsilon_i = \sum_j n_{i,j} / \sum_{i,j} n_{i,j}$.

4.3 Age-Aware Generative Knowledge Distillation

We introduce an age-aware generative knowledge profiling method that mitigates knowledge degradation under DPFL. Specifically, we train a conditional generator G_{ens} to synthesize collective knowledge from both active and idle clients in the knowledge pool $\mathcal{KP}$. Given a latent vector $z \sim \mathcal{N}(0, 1)$ and a class label y sampled from $p(y) = N_y / \sum_{i,j} n_{i,j}$, $y = 1, 2, \ldots, K$, the generator produces class-conditioned samples $\hat{x} = G_{ens}(z, y)$.

To ensure that $\hat{x}$ reflects meaningful and diverse knowledge, we optimize G_{ens} using three loss functions. The first is an age-weighted cross-entropy loss:

$$\mathcal{L}_{ce} = \mathbb{E}_{\hat{x}} \left[\textstyle\sum_{c_i} W_i^y \cdot CE(\theta_i(\hat{x}), y) \right] \tag{5}$$

where weight $W_i^y = aw_i + \frac{n_{i,y}}{N_y}$ accounts for model staleness of c_i on data class y:

Second, to ensure feature-level discriminability, we apply contrastive learning to supervise the feature space via Θ:

$$\mathcal{L}_{ctr} = -\log \frac{exp(d(f(\hat{x}), cen(y))/\tau)}{\sum_{y'} exp(d(f(\hat{x}), cen(y'))/\tau)} \tag{6}$$

where $f(\cdot)$ is the feature extractor of Θ, $cen(y)$ is the semantic center of class y, $d(\cdot)$ is a similarity function (e.g., cosine similarity), y' iterates over all classes, and τ is a temperature parameter.

To prevent model collapse and promote diversity in G_{ens}, we include a data diversity loss term $\mathcal{L}_{div}$, following the design in (Zhang et al. 2022; Zhu, Hong, and Zhou 2021).

To summarize, the overall objective is to optimize G_{ens} with the loss $\mathcal{L} = \gamma_{ce}\mathcal{L}_{ce} + \gamma_{ctr}\mathcal{L}_{ctr} + \gamma_{div}\mathcal{L}_{div}$, where γ_{ce}, γ_{ctr}, and γ_{div} are weighting factors.

After training the generator, we fine-tune Θ by distilling age-weighted knowledge from the ensemble generated by G_{ens} to obtain an improved global model Θ':

$$\hat{y}_{ens} = \textstyle\sum_{c_i} W_i^y \cdot \hat{y}_i \tag{7}$$

$$\mathcal{L}_{ens} = \mathbb{E}_{\hat{x}} \left[KL(\sigma(\hat{y}_{ens}) || \sigma(\hat{y}_{\Theta})) \right] \tag{8}$$

5 Experiment Results

5.1 Benchmarking Setups

We developed an open-source benchmark platform that supports all DPFL scenarios defined in Sec. 3 and benchmarked all 9 FL models reviewed in Sec. 2. We conducted experiments on image classification dataset *Office-Caltech* (Huang et al. 2023). The dataset is partitioned following the settings in Sec. 3. Due to space limits, we present the key results focusing on the most challenging setting, Heavy-NIID Office-Caltech, in the main paper, which is most affected by DP due to high data heterogeneity and limited client overlap. Full experimental results on other datasets and settings are provided on the benchmarking platform.

We adopted the widely used ResNet-10 (He et al. 2016) with a 512-dimensional feature vector for training. Experiments were conducted with 10–50 clients, using a batch size of 128 and running for $T_f = 100$ rounds, each containing 5 local epochs. SGD optimizer was used with a learning rate of 0.01, weight decay of 1×10^{-5}, and momentum of 0.9.

Configurations of the DP patterns are as follows: (i) Timed-random: each client has a joining probability of $p_i(t) = 0.5$. (ii) Markovian: transition probabilities are set as $[p_{0\to0}, p_{0\to1}] = [0.8, 0.2]$ and $[p_{1\to0}, p_{1\to1}] = [0.2, 0.8]$.

5.2 Performance Benchmarking under DPFL

The following results are averaged over 5 runs with different random seeds.

	Metric	Type	FedAvg	FedProx	SCAF.	FedMD	FedGen	MOON	FPL	FLwF	CFeD	AVG
w/o KPFL	$WE(\uparrow)$	Stat.	63.02	64.99	63.20	52.75	73.23	63.07	58.19	65.33	57.98	62.42
		T-R	46.01	47.83	50.03	34.50	56.05	46.40	46.15	46.34	51.74	47.23
		M	50.92	53.30	51.32	33.59	54.75	52.44	51.28	53.22	53.05	50.43
	$IDP(\downarrow)$	T-R	20.87	20.57	17.82	15.80	20.65	21.06	17.95	21.19	11.40	18.59
		M	19.12	18.25	15.80	17.19	21.95	19.11	15.72	19.02	11.62	17.53
	$ID(\downarrow)$	Stat.	0.49	0.46	0.38	0.54	0.29	0.52	0.69	0.43	0.29	0.45
		T-R	1.58	1.74	1.44	1.24	1.72	1.72	1.52	1.68	1.49	1.57
		M	1.11	1.22	1.19	1.02	1.24	1.31	1.03	1.18	1.10	1.16
w/ KPFL	$WE(\uparrow)$	Stat.	66.79	66.06	66.75	53.51	**72.17**	65.59	58.70	65.93	65.46	64.55
		T-R	60.57	62.49	61.75	49.04	**65.00**	61.31	54.06	59.85	63.12	59.69
		M	58.10	59.88	57.79	44.15	**62.66**	60.50	53.26	59.05	58.25	57.07
	$IDP(\downarrow)$	T-R	4.78	4.34	5.02	**1.13**	9.86	5.35	5.31	6.29	4.38	5.16
		M	8.17	6.82	8.63	**5.39**	11.18	7.91	6.30	8.03	8.80	7.91
	$ID(\downarrow)$	T-R	0.52	0.43	0.48	0.47	0.45	**0.36**	0.48	**0.36**	0.47	0.45
		M	0.50	0.45	0.52	**0.36**	0.47	0.41	0.51	0.40	0.47	0.45

Table 1: Benchmarking result on Heavy-NIID Office-Caltech.

Final WE with $\omega = 5$: As shown in Tab. 1, while most FL methods perform reliably under static setting, the introduction of DP severely degrades model performance. Under Timed-Random and Markovian, all methods failed to retain existing knowledge as clients drop out and rejoin, resulting in accuracy losses of around 15% and 12%, respectively.

Intransigence to DP (IDP): All evaluated methods yield positive IDP scores, showing that DP consistently impairs convergence against static baselines. For example, FedAvg yields a score of 20.87 on Timed-Random and 19.12 on Markovian, revealing large convergence gaps caused by DP.

Instability due to DP (ID): Static participation maintains stable learning with an average ID score of 0.45. In contrast, highly dynamic participation, particularly Timed-Random, leads to the highest instability of 1.57. Robustness methods also struggled, FedProx and FedGen reach ID of 1.7 on Timed-Random, and MOON records 1.3 on Markovian.

As shown, DP significantly degrades all existing FL methods across metrics. No existing method demonstrates reliable robustness, highlighting a critical gap in handling DP.

5.3 KPFL Plugin: Effectiveness and Impact

Fully compatible to existing FL models. We successfully integrated KPFL into all 9 FL models reviewed in Sec. 2. Tab. 1 below presents the results of KPFL-enhanced models. Across all DP types, plugging in KPFL consistently improves final accuracy WE, achieving more than 12% gain under Timed-Random. KPFL performs most outstandingly in IDP, narrowing the convergence gap from 18.59→5.16 under Timed-Random and from 17.53→7.91 under Markovian. KPFL also restores stability, reducing the ID scores to the level of static participation.

Scaling KPFL with Larger Client Pools. Tab. 2 reports the scalability of KPFL on top of FedProx. As the client pool increases from 20 to 50, KPFL consistently improves performance across all metrics. While larger client pools exhibit greater stability, KPFL further reinforces learning robustness, demonstrating its scalability to larger client scales.

These results highlight both the limitations of existing methods and the effectiveness of the proposed $\mathcal{KP}$ design in managing varying client availability.

	$WE(\uparrow)$			$IDP(\downarrow)$		$ID(\downarrow)$	
	Stat.	T-R	M	T-R	M	T-R	M
number of clients = 20							
FedProx	59.48	51.50	45.50	13.64	11.63	1.15	1.48
+KPFL	**60.58**	**57.41**	**54.13**	**4.23**	**9.20**	**0.39**	**0.49**
number of clients = 50							
FedProx	51.41	45.20	41.50	9.17	8.93	1.12	0.99
+KPFL	**53.01**	**50.30**	**48.26**	**3.59**	**7.31**	**0.29**	**0.42**

Table 2: Performance gains of KPFL-enhanced FedProx under Heavy-NIID Office-Caltech.

		FedProx +			MOON +		
		MIFA	KPFL		MIFA	KPFL	
			Θ	Θ'		Θ	Θ'
$WE(\uparrow)$	Stat.	64.04	65.79	**66.06**	63.57	65.29	**65.59**
	T-R	60.55	60.28	**62.49**	59.15	60.98	**61.31**
	M	**61.03**	57.89	59.88	59.71	57.55	**60.50**
$ID(\downarrow)$	T-R	0.46	**0.42**	0.43	0.51	0.47	**0.36**
	M	0.40	**0.38**	0.45	0.42	0.46	**0.41**

Table 3: Ablation study on Θ and Θ' (Heavy-NIID Office-Caltech). MIFA is included as an alternative aging approach.

5.4 Ablation Study

In KPFL, there is an initial global model Θ and an improved Θ'. Tab. 3 shows our ablation study results. MIFA (Gu et al. 2021) is included as a baseline, as it also stores the latest gradients for idle clients but applies static weights. In contrast, KPFL employs a dual-age strategy, which explicitly takes idle and active clients' states into account. The results show that Θ' consistently outperforms Θ, validating the benefit of our two-stage design. Additionally, comparison with Tab. 1 highlights the individual contributions and overall efficacy of KPFL. Furthermore, Θ' outperforms MIFA in most cases, showing substantial improvements in learning robustness, which demonstrates the effectiveness of our design.

6 Conclusions

This paper presents several benchmarking scenarios to systematically investigate the challenges of DPFL. Our study demonstrates that DP significantly impacts the learning efficiency of a wide range of existing FL models across diverse datasets. Furthermore, the proposed KPFL plugin effectively mitigates the effects of participation dynamics across various FL model types.

Acknowledgements

This work was supported by grants of National Science and Technology Council (NSTC 114-2221-E-A49-11), and the NVIDIA Academic Grant Program 2025.

References

Beutel, D. J.; Topal, T.; Mathur, A.; Qiu, X.; Fernandez-Marques, J.; Gao, Y.; Sani, L.; Li, K. H.; Parcollet, T.; de Gusmão, P. P. B.; et al. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Du, Z.; Wu, C.; Yoshinaga, T.; Yau, K.-L. A.; Ji, Y.; and Li, J. 2020. Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues. *IEEE Open Journal of the Computer Society*, 1: 45–61.

Düsing, C.; and Cimiano, P. 2025. Rethinking federated learning as a digital platform for dynamic and value-driven participation. *Future Generation Computer Systems*, 171: 107847.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv preprint arXiv:1312.6211*.

Gu, X.; Huang, K.; Zhang, J.; and Huang, L. 2021. Fast Federated Learning in the Presence of Arbitrary Device Unavailability. In *Advances in Neural Information Processing Systems*.

He, C.; Li, S.; So, J.; Zeng, X.; Zhang, M.; Wang, H.; Wang, X.; Vepakomma, P.; Singh, A.; Qiu, H.; et al. 2020. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, W.; Ye, M.; Shi, Z.; Li, H.; and Du, B. 2023. Rethinking federated learning with domain shift: A prototype view. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. of International Conference on Machine Learning (ICML)*.

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

Lan, L.; Wang, J.; Li, Z.; Kant, K.; and Liu, W. 2024. FedREM: Guided Federated Learning in the Presence of Dynamic Device Unpredictability. *IEEE Transactions on Parallel and Distributed Systems*, 35(7): 1189–1206.

Li, D.; and Wang, J. 2019. FedMD: Heterogenous Federated Learning via Model Distillation. *arXiv preprint arXiv:1910.03581*.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated learning on Non-IID data silos: An experimental study. In *Proc. of International Conference on Data Engineering (ICDE)*.

Li, Q.; He, B.; and Song, D. 2021. Model-contrastive federated learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*.

Liang, J.-M.; Chang, P.-Y.; Chen, J.-J.; Huang, C.-F.; and Tseng, Y.-C. 2018. Energy-efficient DRX scheduling for D2D communication in 5G networks. *Journal of Network and Computer Applications*, 116: 53–64.

Liang, J.-M.; Chen, J.-J.; Hsieh, P.-C.; and Tseng, Y.-C. 2016. Two-Phase Multicast DRX Scheduling for 3GPP LTE-Advanced Networks. *IEEE Transactions on Mobile Computing*, 15(7): 1839–1849.

Liang, J.-M.; Hsu, C.-K.; Chen, J.-J.; Wu, K.-R.; and Tseng, Y.-C. 2020. Three-Stage DRX Scheduling for Joint Downlink Transmission in C-RAN. *IEEE Wireless Communications Letters*, 9(2): 129–133.

Ma, Y.; Xie, Z.; Wang, J.; Chen, K.; and Shou, L. 2022. Continual Federated Learning Based on Knowledge Distillation. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. 24: 109–165.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. of International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Ruan, Y.; Zhang, X.; Liang, S.-C.; and Joe-Wong, C. 2021. Towards flexible device participation in federated learning. In *Proc. of International Conference on Artificial Intelligence and Statistics (AISTATS)*.

TensorFlow Federated Authors. 2019. TensorFlow Federated. https://www.tensorflow.org/federated.

Usmanova, A.; Portet, F.; Lalanda, P.; and Vega, G. 2021. A distillation-based approach integrating continual learning and federated learning for pervasive services. *arXiv preprint arXiv:2109.04197*.

Xu, C.; Qu, Y.; Xiang, Y.; and Gao, L. 2023. Asynchronous federated learning on heterogeneous devices: A survey. *Computer Science Review*, 50: 100595.

Yang, X.; Yu, H.; Gao, X.; Wang, H.; Zhang, J.; and Li, T. 2024. Federated continual learning via knowledge fusion: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, L.; Shen, L.; Ding, L.; Tao, D.; and Duan, L.-Y. 2022. Fine-tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, T.; Gao, L.; Lee, S.; Zhang, M.; and Avestimehr, S. 2023. TimelyFL: Heterogeneity-Aware Asynchronous Federated Learning With Adaptive Partial Training. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Zhou, Y.; Pang, X.; Wang, Z.; Hu, J.; Sun, P.; and Ren, K. 2024. Towards Efficient Asynchronous Federated Learning in Heterogeneous Edge Environments. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*.

Zhu, Z.; Hong, J.; and Zhou, J. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *Proc. of International Conference on Machine Learning (ICML)*.

Ziller, A.; Trask, A.; Lopardo, A.; Szymkow, B.; Wagner, B.; Bluemke, E.; Nounahon, J.-M.; Passerat-Palmbach, J.; Prakash, K.; Rose, N.; et al. 2021. Pysyft: A library for easy federated learning. *Federated learning systems: Towards next-generation AI*, 111–139.