nature communications

Article

https://doi.org/10.1038/s41467-025-58055-3

Towards fairness-aware and privacy-preserving enhanced collaborative learning for healthcare

Received: 26 February 2024

Accepted: 11 March 2025

Published online: 23 March 2025

Check for updates

Feilong Zhang[1,5], Deming Zhai[1,5], Guo Bai[2], Junjun Jiang[1], Qixiang Ye[3], Xiangyang Ji[4] & Xianming Liu[1]

The widespread integration of AI algorithms in healthcare has sparked ethical concerns, particularly regarding privacy and fairness. Federated Learning (FL) offers a promising solution to learn from a broad spectrum of patient data without directly accessing individual records, enhancing privacy while facilitating knowledge sharing across distributed data sources. However, healthcare institutions face significant variations in access to crucial computing resources, with resource budgets often linked to demographic and socioeconomic factors, exacerbating unfairness in participation. While heterogeneous federated learning methods allow healthcare institutions with varying computational capacities to collaborate, they fail to address the performance gap between resource-limited and resource-rich institutions. As a result, resource-limited institutions may receive suboptimal models, further reinforcing disparities in AI-driven healthcare outcomes. Here, we propose a resource-adaptive framework for collaborative learning that dynamically adjusts to varying computational capacities, ensuring fair participation. Our approach enhances model accuracy, safeguards patient privacy, and promotes equitable access to trustworthy and efficient AI-driven healthcare solutions.

The increasing integration of AI algorithms in medicine and healthcare has raised significant ethical concerns, with privacy and fairness at the forefront[1–4]. As these algorithms leverage vast amounts of sensitive patient data for training and inference, there is a heightened risk of privacy breaches and unauthorized access[5]. Safeguarding patient privacy becomes crucial to maintain trust in medicine and healthcare systems[6]. Additionally, concerns about fairness arise due to the potential biases in AI algorithms, especially when trained on datasets that may not be representative of diverse demographics[7,8]. Biased algorithms can lead to unequal treatment and diagnostic inaccuracies, disproportionately affecting certain patient groups[9,10]. Achieving a balance between the advantages of AI in medicine and healthcare and alleviating public concerns regarding ethical issues requires ongoing efforts to tackle privacy protection and biases in algorithmic decision-making. This underscores the ethical responsibility of stakeholders in the medicine and healthcare, as well as technology sectors.

Federated Learning (FL) has emerged as a groundbreaking collaborative learning approach to address the delicate balance between data sharing and privacy concerns[11–15]. This innovative technique enables multiple institutions or devices to collaboratively train machine learning models without the need to centrally pool sensitive data. The practice of keeping data localized and conducting model updates on-site significantly reduces the risks associated with centralized data storage and processing. The decentralized nature of FL aligns seamlessly with stringent data protection regulations, such as the Health Insurance Portability and Accountability Act (HIPAA)[16] in the

[1]Faculty of Computing, Harbin Institute of Technology, Harbin, China. [2]Shanghai Ninth People's Hospital, School of Medicine, Shanghai Jiao Tong University, Shanghai, China. [3]School of Electronic Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. [4]Department of Automation, Tsinghua University, Beijing, China. [5]These authors contributed equally: Feilong Zhang, Deming Zhai. e-mail: xyji@tsinghua.edu.cn; csxm@hit.edu.cn

United States or the General Data Protection Regulation (GDPR)[17] in Europe. Moreover, FL serves as a promising approach to promoting algorithmic fairness in AI for medicine and healthcare. The decentralized training process of FL enables the inclusion of diverse datasets from various healthcare institutions and demographics, contributing to a more representative and equitable model. As the federated model learns from a broad spectrum of patient data without directly accessing individual records, the potential for bias is reduced, leading to fairer and more robust models[18–20].

However, despite its promise, FL faces critical challenges related to computational resource diversity, which introduces implicit fairness issues in participation. Healthcare institutions vary significantly in computational capabilities, with resource-constrained organizations such as hospitals in developing countries often lacking the infrastructure to handle the substantial demands of large-scale FL[12]. In traditional model-homogeneous FL, where all clients train on the same large model, powerful clients benefit more because they can fully leverage their resources, while weaker clients are often excluded from participation due to their limited computational capacity. Conversely, when smaller models are used to accommodate weaker clients, the computational resources of stronger clients are underutilized, leading to a trade-off that undermines both fairness and efficiency. Even in heterogeneous FL methods, which aim to allow clients with varying resource levels to participate, significant performance disparities between strong and weak clients persist. Stronger clients typically achieve much higher model accuracy than weaker ones, creating a severe fairness issue that exacerbates existing healthcare inequalities. Addressing this problem is critical to ensuring that the benefits of FL are equitably distributed across healthcare providers with diverse resource capabilities.

We claim that, a fair and sustainable collaborative learning system should satisfy the following five key principles: 1) Equal Opportunity: All collaborators, irrespective of their resource strength, should have an equal opportunity to participate in the collaborative learning process. This includes equal opportunities for model updates and participation in communication. This helps avoid inequalities and biases between nodes, ensuring that each node's voice is fully considered. 2) Fair Contribution: The FL system should encourage and accommodate contributions from all nodes. Even if nodes have different scales or types of data as well as computational resources, their contributions should be considered valuable. Technically, this might involve mechanisms to ensure that the contribution of each node influences the global model. 3) Shared Fruits: During the model training process, ensure that the learned global model is available to each node, and contributions from each node are equally reflected in the overall performance during the testing phase. This can be achieved by considering the impact of each node when aggregating the model. 4) Equal Model Test Accuracy: Emphasize that all nodes should have the same model test accuracy to prevent bias and unfairness resulting from unequal treatment. This may require the consideration of fairness metrics in the design of FL algorithms and incorporating these metrics during model evaluation and aggregation. 5) Sustainability: The system should be designed to be sustainable, ensuring that FL can continue in the face of dynamic changes in node participation, resource availability, and engagement levels. This might include smooth transition mechanisms when nodes dynamically join or leave and adaptive strategies for situations where node resources are insufficient.

From a technical perspective, the resource heterogeneity across different participating clients involves variations in processing power, memory and communication capabilities[21]. Ensuring effective coordination and resource allocation becomes intricate when dealing with such diversity. Despite this fact, traditional FL paradigms, such as FedAvg[11], adhere to the uniform model capacity assumption, where all participating clients share the same model architecture. In the era of large language models (LLM), training large models seems like an inevitable choice. A widely-adopted strategy to match this assumption is to employ a large-scale model as the uniform global model for collaborative training, which, however, results in the following challenges:

- Unfairness Issue. The utilization of a uniform large global model poses a significant barrier for clients with constrained computational resources, making their participation in collaborative training unfeasible. This exclusion is problematic as these clients may struggle to bear the costs associated with local training efforts. Resource budgets are often intertwined with the demographic and socio-economic status of the owners, exacerbating the issue of unfairness in terms of participation. The omission of such resource-constrained clients not only introduces training bias due to the absence of their data perspective but also exacerbates overall fairness concerns. Additionally, during the deployment phase, participating clients equipped with limited computational resources face challenges in accommodating large-scale models, further underscoring the need for a more resource-sensitive approach in the FL paradigm.
- Huge Communication Overhead. During the training phase, intensive communication between the server and participating clients occurs for exchanging local model updates across multiple rounds. The prohibitive cost of transmitting parameters, particularly with a large global model, emerges as a significant bottleneck in the training process[22]. This limitation not only hinders the participation of clients with weaker communication conditions in collaborative learning but also leads to substantial energy consumption, rendering the training procedure practically infeasible and environmentally unfriendly.
- Fragile Privacy Protection. The uniform model architecture introduces a vulnerability in terms of privacy protection. When coordinating model updates between the server and participating clients, a malicious server can exploit this uniformity to infer local data by analyzing stolen gradients–a threat commonly known as gradient inversion attack[23]. This poses a considerable risk to the security and privacy preservation of FL.

The successful deployment of FL for medicine and healthcare in diverse computing resources hinges on overcoming these challenges to unlock the full potential of decentralized, privacy-preserving machine learning. A scheme that embodies fairness, communication-efficient characteristics, and enhanced privacy protection becomes essential for building a trustworthy and effective medicine and healthcare AI ecosystem that prioritizes patient well-being and confidentiality.

To address the heterogeneity of participating clients, it becomes imperative to develop an FL framework capable of accommodating a spectrum of diverse local models. Such a framework must exhibit adaptability to varying neural network complexities in accordance with computational resource budgets. However, this introduces a new challenge: How can knowledge be exchanged effectively across these heterogeneous local models to derive a unified global model? In the literature, some strategies tailored for heterogeneous FL have been explored, with notable works adopting either knowledge distillation-based approaches[24–27] or network pruning-based techniques[28–30]. Knowledge distillation allows edge clients to train diverse local models, which are subsequently distilled into a uniform global model for aggregation. However, a well-known limitation is the inability of knowledge distillation to losslessly transfer knowledge across diverse network structures, resulting in a loss of model accuracy[31]. On the other hand, pruning-based approaches modulate the size of Convolutional Neural Networks (CNNs) by adjusting the width or depth of networks, offering a means to adapt model capacity. Yet, disparities in behavior between sub-networks and complete networks can lead to a mismatch of feature spaces.

https://doi.org/10.1038/s41467-025-58055-3

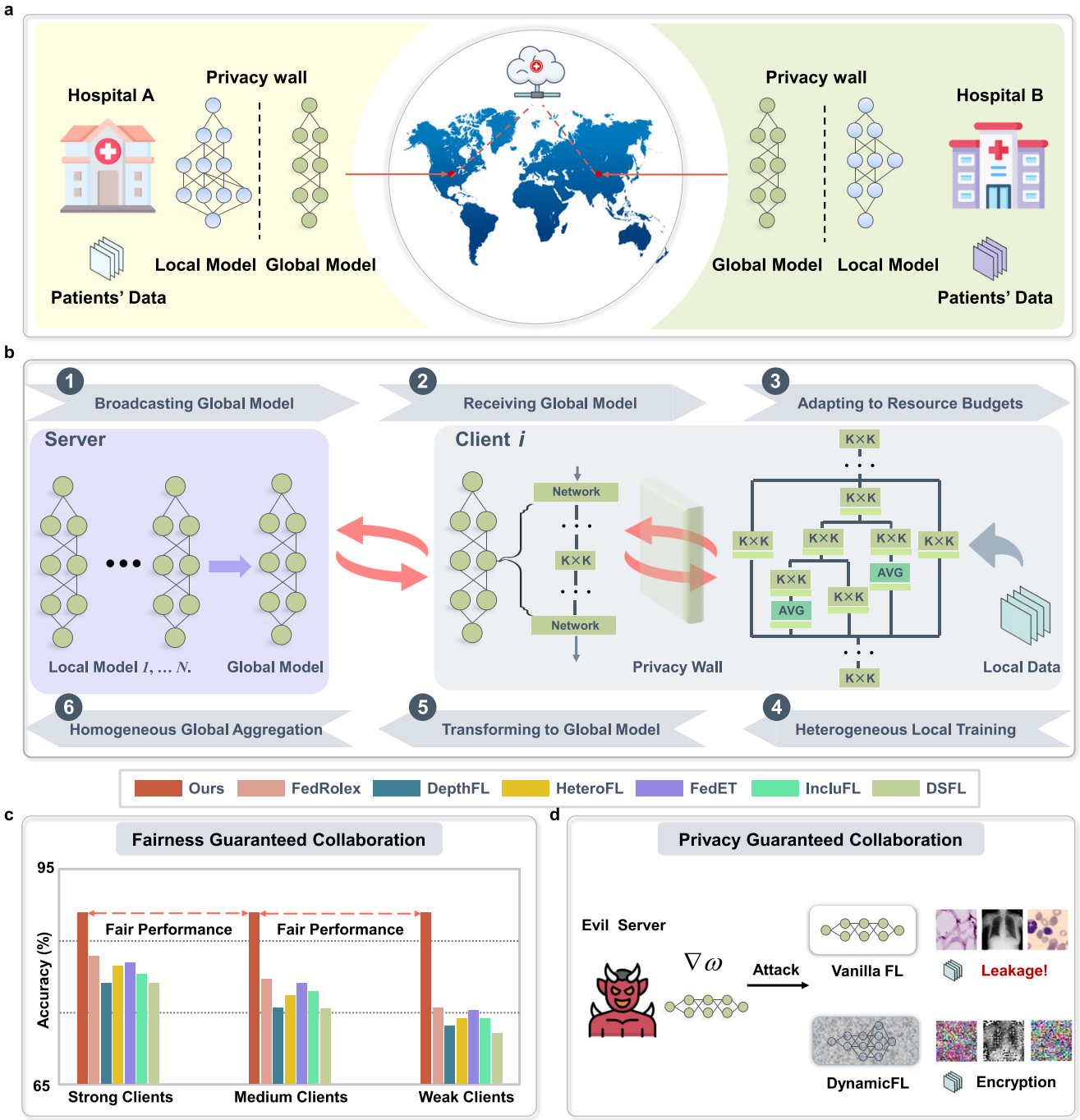

Fig. 1 | The overall framework of DynamicFL. a, Healthcare institutions in developing and developed countries exhibit significant variations in accessing crucial computing resources, with resource budgets often linked to demographic and socio-economic factors, exacerbating unfairness in participation of collaborative learning for medicine and healthcare. **b**, The workflow of the proposed DynamicFL scheme. **c**, Fairness guaranteed collaboration: Comparison of test accuracy across clients with varying computational capabilities under different FL algorithms. Each column represents a distinct FL method, with the order of bars corresponding to the legend above for clarity. DynamicFL ensures consistent test accuracy across clients, fostering fairness in the distribution of training outcomes. **d**, Privacy guaranteed collaboration: DynamicFL can effectively counteract gradient inversion attacks, achieving enhanced privacy preservation capabilities, even in scenarios involving gradient leakage. Source data are provided as a Source Data file.

While knowledge distillation and network pruning are popular strategies, they suffer from limitations such as lossy knowledge transfer and feature space mismatch. Consequently, exploring an alternative approach to unlock the full potential of all clients becomes imperative. This paper introduces an effective approach to heterogeneous FL through a unified framework. Specifically, we propose the Dynamic Federated Learning (DynamicFL) framework, which ensures fair participation of all clients in collaborative learning, allowing them to work at full capacity and ensuring lossless knowledge transfer across diverse local models. The overall process is illustrated in Fig. 1. The motivation of our scheme involves a thoughtful examination of two pivotal issues: 1) designing a resource-adaptive network architecture for each client to harness their complete strength and 2) projecting heterogeneous local models into a uniform model architecture to streamline the aggregation operation. In response to these considerations, the key components of DynamicFL are tailored,

encompassing heterogeneous local training and homogeneous global aggregation:

– Heterogeneous Local Training: Customizing the training approach for each client, DynamicFL dynamically expands local models to multi-branches based on the local computational resources, as shown in Fig. 1 (b). This expansion is achieved through re-parameterization, drawing capacity from the VGG-style plain global model. Each client operates on its modulated local model at full computational capacity, ensuring high accuracy in local training. A noteworthy distinction from conventional structural re-parameterization, such as those seen in[32,33], lies in our dynamic adjustment of computation workloads. This involves selecting operations with significant contributions to performance, striking a balance between training accuracy and efficiency.

– Homogeneous Global Aggregation: After local training, the heterogeneous local models undergo re-parameterization to transform back to the original global model structure. This ensures that all uploaded local models share the same structure, facilitating aggregation operations without incurring knowledge transfer overhead. The absence of intricate knowledge distillation on the server side streamlines the aggregation process, allowing for rapid execution.

Our proposed approach not only guarantees fair participation across diverse institutions, regardless of their computational strength, but also minimizes communication overhead. Furthermore, it strengthens the adherence to security and privacy standards, essential for building and sustaining trust in AI applications within the medicine and healthcare domain.

Results

In this section, we present extensive experimental results to illustrate the superiority of DynamicFL in terms of prediction accuracy, convergence speed, test accuracy variance (i.e., performance fairness among clients), and robustness to gradient attacks. We also demonstrate the advantage of DynamicFL as a plug-and-play module for enhancing the performance of existing methods.

Comparison Baselines

To evaluate the effectiveness of our proposed method, we compare it against several widely recognized and highly cited baselines spanning different categories of FL approaches. For knowledge distillation-based methods, we include DSFL[26], FedET[27], IncluFL[34], FedMD[35] and FedDF[25], which aim to achieve communication efficiency and model personalization through knowledge distillation. For network pruning-based methods, we evaluate HeteroFL[28], DepthFL[36], and FedRolex[37], which focus on adapting model complexity to clients' heterogeneous resources. Furthermore, we extend the comparison to include three recently proposed methods, FCCL[38], pFedHR[39] and FedTGP[40], to benchmark our approach against the latest advancements in the field. It is worth noting that most of the original implementations of the compared methods did not include experiments on ViT, and many of them cannot be directly applied to ViT without significant modifications. Therefore, for the ViT-based experiments, we limit the baselines to knowledge distillation-based methods such as FedDF and FedMD, which can naturally scale to ViT.

Experiment Setup

Datasets. In our experiments, we conducted a thorough evaluation of DynamicFL's performance using two well-known benchmark datasets, CIFAR-10 and CIFAR-100, as well as three specialized medical image datasets: CancerSlides[41,42], ChestXray[43,44], and BloodCell[45]. Specifically, the CancerSlides dataset focuses on the prediction of survival outcomes in colorectal cancer, featuring 10,000 non-overlapping image patches extracted from hematoxylin & eosin-stained histological slides. Additionally, it includes a test set of 7180 patches sourced from a different clinical center. The ChestXray dataset consists of 3616 COVID-19 positive cases along with 10,192 normal, 6012 lung opacity (Non-COVID lung infection), and 1345 viral pneumonia images. Lastly, the BloodCell dataset comprises 17,092 images of single normal cells, collected from individuals without infection, hematologic or oncological diseases, and not under any pharmacological treatment at the time of blood draw. This dataset categorizes images into 8 distinct classes, facilitating a detailed analysis and classification. To align with the input requirements of different architectures, we resized all datasets to 32 × 32 for experiments involving convolutional neural networks (CNNs) and to 224 × 224 for transformer-based models.

Data & Clients Heterogeneity. To simulate statistical heterogeneity, we employ the Dirichlet distribution $p_k \sim Dir_N(\beta)$ to construct Non-IID data partitions across clients. To simulate the heterogeneous device scenario, clients are categorized into three types—`weak`, `medium`, and `strong` clients. This classification is grounded on the capability of each client to run models of varying sizes.

Experimental Settings. For CNN models, the learning rate is set to 0.01, while for transformer models, the learning rate is set to 0.001. The batch size for all experiments is set to 16, except for ViT-1B, where the batch size is set to 3 due to the model's large size. Unless otherwise specified, the number of clients in all experiments is 30, except for ViT-1B, which involves 9 clients. Our experiments are implemented using PyTorch and tested on various platforms, including several machines with 4 NVIDIA 3090 GPUs, several machines with 4 NVIDIA 4090 GPUs, and a machine with a single NVIDIA A100 GPU.

Evaluation Strategy

In our approach, we adapt the global model to more complex architectures, allowing for the accommodation of varying computational capabilities among the diverse clients participating in the FL process. Specifically, our global model is designed to be equivalent to the smallest local model, which corresponds to the weakest client. This strategy enables us to develop a lightweight neural network through FL. For a fair comparison, the knowledge distillation-based methods we compare against use the same local models as our method. However, their global model is the largest local model, corresponding to the strongest client. This distinction results in a significantly larger global model for the knowledge distillation-based methods compared to ours. The comparison with pruning-based methods presents a different challenge. Achieving identical local models as our method is difficult with pruning-based approaches. We align these methods' global models with those used in the distillation-based methods, while employing pruned versions of these models as their local models. Consequently, both the knowledge distillation and pruning-based methods utilize a much larger global model than our approach. To evaluate the effectiveness of the compared FL methods, we focus on the accuracy of their global models on the test datasets.

Performance Evaluation Under Equal Wall-Clock Time

We evaluate the performance of DynamicFL compared to several baseline methods on three medical datasets (CancerSlides, ChestXray, and BloodCell) under the same wall-clock training time. The experiments are conducted under both IID and Non-IID data distributions, where Non-IID heterogeneity is introduced using a Dirichlet distribution Dir(β) with a concentration parameter $\beta = 0.5$. To simulate realistic heterogeneous environments, clients are divided into strong, medium, and weak categories in an equal ratio of 1:1:1. Each experiment is repeated five times to ensure statistical reliability, and results are reported as mean accuracy ± standard deviation, with the best-performing results highlighted in bold.

As shown in Table 1, DynamicFL consistently outperforms all baseline methods across the three datasets under both IID and Non-IID settings. On the CancerSlides dataset, DynamicFL achieves an accuracy of 87.37% under IID settings, outperforming the strongest baseline

Table 1 | Performance evaluation of DynamicFL on three medical datasets under the same wall-clock training time

	CancerSlides[41,42]		ChestXray[43]		BloodCell[45]	
Methods	**IID**	**Non-IID**	**IID**	**Non-IID**	**IID**	**Non-IID**
DSFL[26]	78.58 ± 1.87	70.79 ± 2.30	81.04 ± 2.33	73.85 ± 1.97	78.47 ± 2.12	72.11 ± 2.08
FedET[27]	80.39 ± 1.58	71.25 ± 1.74	82.13 ± 1.32	76.24 ± 1.68	81.79 ± 1.74	74.96 ± 1.91
IncluFL[34]	81.26 ± 2.73	67.83 ± 2.39	80.69 ± 2.27	74.31 ± 2.44	80.36 ± 2.52	73.82 ± 2.65
FedMD[35]	78.67 ± 1.94	70.18 ± 2.12	80.13 ± 1.87	72.93 ± 2.05	78.91 ± 2.07	71.54 ± 2.21
FedTGP[40]	81.92 ± 1.63	73.69 ± 1.86	82.14 ± 1.44	75.92 ± 1.78	81.39 ± 1.82	74.39 ± 2.03
FedDF[25]	78.82 ± 1.88	70.04 ± 2.05	80.22 ± 2.02	73.01 ± 1.86	79.04 ± 2.09	71.78 ± 2.18
pFedHR[39]	81.54 ± 1.77	74.12 ± 2.03	82.34 ± 1.59	76.46 ± 1.92	81.73 ± 2.01	74.79 ± 2.14
FCCL[38]	81.83 ± 1.89	74.65 ± 2.17	82.52 ± 1.72	76.75 ± 2.14	82.06 ± 2.05	75.41 ± 2.26
HeteroFL[28]	79.17 ± 2.31	68.34 ± 2.57	81.92 ± 2.16	73.60 ± 2.44	81.19 ± 2.69	72.25 ± 2.88
DepthFL[36]	75.49 ± 2.36	62.84 ± 2.92	77.61 ± 1.88	69.84 ± 2.24	78.77 ± 1.94	68.85 ± 2.31
FedRolex[37]	81.58 ± 1.17	68.76 ± 1.49	82.46 ± 1.36	74.22 ± 1.48	82.73 ± 1.71	72.18 ± 1.52
DynamicFL	**87.37 ± 0.86**	**80.26 ± 0.57**	**87.92 ± 0.79**	**83.41 ± 0.92**	**88.64 ± 0.44**	**81.76 ± 0.71**

The table presents the performance evaluation of DynamicFL and baseline methods on three medical datasets (CancerSlides, ChestXray, and BloodCell) under both IID and Non-IID settings. The results in bold indicate the best performance. Non-IID heterogeneity is introduced using the Dirichlet distribution Dir(β) with a concentration parameter $\beta = 0.5$. The experiments are conducted with an equal ratio of strong, medium, and weak clients (1:1:1), and each experiment is repeated five times to ensure statistical reliability. The results are reported as the mean accuracy ± standard deviation. DynamicFL consistently outperforms other methods, as indicated by the bolded results, demonstrating its robustness and effectiveness in both IID and Non-IID scenarios. Source data are provided as a Source Data file. Source data are provided as a Source Data file.

Table 2 | Performance evaluation of DynamicFL and other methods on ResNet-18 and ViT-Base architectures

Model	Methods	Ratio (7:2:1)	Ratio (5:2:3)	Ratio (4:1:5)	Ratio (4:3:3)	Ratio (3:6:1)
ResNet-18	DSFL[26]	70.12	73.34	68.45	74.23	71.56
	FedET[27]	71.78	74.56	69.78	75.45	73.12
	IncluFL[34]	72.45	75.12	70.45	76.12	73.78
	FedMD[35]	70.89	73.01	68.12	74.01	71.67
	FedTGP[40]	74.56	76.23	74.45	75.12	73.34
	FedDF[25]	73.34	76.01	71.78	76.89	74.23
	pFedHR[39]	74.45	77.34	73.89	77.45	75.67
	FCCL[38]	73.78	76.12	74.78	77.01	73.89
	HeteroFL[28]	72.34	75.45	70.56	76.34	73.12
	DepthFL[36]	68.45	71.12	66.12	72.34	69.78
	FedRolex[37]	73.89	76.23	71.45	77.01	74.45
	DynamicFL	**82.12**	**83.34**	**80.12**	**83.45**	**81.23**
ViT-Base	DSFL[26]	69.34	72.12	67.45	73.01	70.45
	FedET[27]	70.89	73.34	68.78	74.12	71.67
	IncluFL[34]	71.56	74.12	69.45	74.89	72.34
	FedMD[35]	70.12	72.89	67.78	73.34	70.89
	FedDF[25]	72.45	75.12	70.34	75.89	73.12
	DynamicFL	**80.45**	**81.34**	**78.89**	**82.45**	**79.67**

The table presents the performance of DynamicFL under different client distribution ratios: (7:2:1), (5:2:3), (4:1:5), (4:3:3), and (3:6:1). These ratios represent the proportion of data distributed among clients with varying computational and data resources, reflecting heterogeneous federated learning scenarios. The results highlight that DynamicFL consistently outperforms baseline methods across all distribution ratios, demonstrating its ability to handle heterogeneous client distributions effectively and achieve superior performance in both ResNet-18 and ViT-Base architectures. The results in bold indicate the best performance. Source data are provided as a Source Data file.

FCCL by 5.54%, while under Non-IID settings, it achieves 80.26%, surpassing FCCL's 74.65% by a significant margin. For the ChestXray dataset, DynamicFL achieves an accuracy of 87.92% under IID settings, which is 5.40% higher than FCCL's 82.52%, and under Non-IID settings, it reaches 83.41%, significantly outperforming pFedHR's 76.46% by 6.95%. On the BloodCell dataset, DynamicFL achieves an accuracy of 88.64% under IID settings, which is 5.91% higher than FedRolex's 82.73%, and under Non-IID settings, it achieves 81.76%, outperforming FCCL's 75.41% by 6.35%.

These results demonstrate that DynamicFL is highly effective across datasets and consistently outperforms baseline methods in both IID and Non-IID scenarios. Its performance under Non-IID settings is particularly noteworthy, as it demonstrates superior robustness and adaptability to data heterogeneity, making it a strong solution for federated learning in realistic, heterogeneous environments.

Performance Evaluation Under Diverse Client Distributions

We conduct experiments with client distribution ratios: 7:2:1, 5:2:3, 4:1:5, 4:3:3, and 3:6:1. The results are summarized in Table 2, demonstrating that DynamicFL adapts effectively to diverse client distributions while maintaining superior performance and fairness across all configurations. In the most heterogeneous distribution (7:2:1), DynamicFL achieves 82.12% accuracy on ResNet-18, significantly surpassing the best baseline method (pFedHR, 74.45%) by 7.67%. In the balanced distribution (4:3:3), DynamicFL maintains its superiority with an accuracy of 83.45%, outperforming the best baseline (FCCL, 77.01%) by 6.44%. These results further underline the robustness of DynamicFL across varying levels of client heterogeneity, reinforcing its ability to adapt to real-world FL scenarios.

Performance Evaluation Across Diverse Network Architectures

Generalization on CNN Architectures. We systematically evaluate the performance of DynamicFL on different CNN network architectures, including LeNet-5, GoogLeNet, and ResNet-18, with ratios of 1:2:3. As shown in Table 3, DynamicFL consistently outperforms both distillation-based and pruning-based methods across these diverse network structures. For example, on the BloodCell dataset using LeNet-5, DynamicFL achieves an accuracy of 66.44%, surpassing the pruning-based method FedRolex (60.13%) and the distillation-based method (61.75%). This demonstrates the ability of DynamicFL to adapt to smaller network architectures while maintaining superior performance. Furthermore, by adopting a smaller network as the global model, our approach reduces communication costs compared to pruning-based methods, making it more practical for resource-constrained scenarios such as mobile devices.

Generalization on Transformer Architectures. To further validate the scalability and generalizability of DynamicFL, we conduct experiments on transformer-based models, including ViT-Tiny, ViT-Small, ViT-Base, and ViT-1B, across two datasets: BloodCell and CancerSlides. These experiments are performed under both IID and non-IID settings, where non-IID heterogeneity is introduced using a Dirichlet distribution with a

Table 3 | Performance evaluation of test accuracy (%) with different network structures

	CIFAR-10		CIFAR-100		BloodCell[45]		
LeNet-5	**IID**	**Non-IID**	**IID**	**Non-IID**	**IID**	**Non-IID**	**Global Model Parameters (Communication Overhead)**
DSFL[26]	69.88	62.79	51.12	42.94	69.87	60.07	1.8 M
FedET[27]	75.81	70.29	52.43	46.61	71.53	62.96	1.8 M
IncluFL[34]	74.72	68.46	51.88	47.57	73.49	65.82	1.8 M
HeteroFL[28]	71.39	65.26	50.67	43.71	71.94	63.44	1.8 M
DepthFL[36]	70.22	63.57	49.17	40.15	70.48	63.17	1.8 M
FedRolex[37]	74.72	69.75	52.81	47.38	72.54	65.48	1.8 M
DynamicFL	**79.33**	**74.21**	**63.08**	**55.92**	**77.36**	**71.29**	**0.6 M**
	CIFAR-10		**CIFAR-100**		**BloodCell[45]**		
GoogLeNet	**IID**	**Non-IID**	**IID**	**Non-IID**	**IID**	**Non-IID**	**Global Model Parameters (Communication Overhead)**
DSFL[26]	70.35	64.57	52.61	44.73	70.49	65.37	20.7 M
FedET[27]	72.39	66.64	54.69	48.42	73.18	66.19	20.7 M
IncluFL[34]	72.73	67.86	55.28	46.59	75.29	67.94	20.7 M
HeteroFL[28]	72.93	65.84	52.74	43.19	74.26	67.41	20.7 M
DepthFL[36]	71.12	63.65	50.49	42.97	73.17	66.92	20.7 M
FedRolex[37]	77.85	72.46	54.28	48.75	76.38	70.48	20.7 M
DynamicFL	**82.84**	**77.16**	**61.52**	**54.28**	**81.49**	**75.13**	**6.9 M**
	CIFAR-10		**CIFAR-100**		**BloodCell[45]**		
ResNet-18	**IID**	**Non-IID**	**IID**	**Non-IID**	**IID**	**Non-IID**	**Global Model Parameters (Communication Overhead)**
DSFL[26]	71.31	66.90	53.64	44.97	78.47	72.11	35.1 M
FedET[27]	75.13	69.13	54.37	46.55	81.79	74.96	35.1 M
IncluFL[34]	79.87	72.66	56.72	50.18	80.36	73.82	35.1 M
HeteroFL[28]	74.54	67.95	53.81	45.27	81.19	72.25	35.1 M
DepthFL[36]	75.21	66.78	52.27	44.62	78.77	68.85	35.1 M
FedRolex[37]	80.63	74.75	55.26	50.07	82.73	72.18	35.1 M
DynamicFL	**86.82**	**81.14**	**63.27**	**57.20**	**88.64**	**81.76**	**11.7 M**

Compared method are run under the same wall-clock training time.
The table shows the results of DynamicFL under three different network structures. Non-IID heterogeneity is introduced using the Dirichlet distribution Dir(β) with a concentration parameter $\beta = 1$.The global round number for our method is fixed at 100. And the global round of the other baselines is dependent on their respective computation time, ensuring a fair total computation time compared to our method. The results highlighted in bold indicate superior performance. Source data are provided as a Source Data file.

Table 4 | Performance evaluation of transformer architecture

		Client's Parameters			Accuracy (%)	
Dataset	**Model**	**Weak**	**Medium**	**Strong**	**IID**	**Dirichlet (0.7)**
BloodCell	ViT-Tiny	5.5M	9.1M	12.7M	73.11	65.39
	ViT-Small	21.7M	35.9M	50.1M	82.52	76.09
	ViT-Base	85.8M	142.5M	199.3M	87.02	85.09
	ViT-1B	1.3B	2.9B	7.1B	94.77	91.26
CancerSlides	ViT-Tiny	5.5M	9.1M	12.7M	79.64	77.45
	ViT-Small	21.7M	35.9M	50.1M	83.21	81.34
	ViT-Base	85.8M	142.5M	199.3M	85.52	82.42
	ViT-1B	1.3B	2.9B	7.1B	92.64	89.53

The table presents the performance of different sizes of Vision Transformers (ViT), including ViT-Tiny, ViT-Small, ViT-Base, and ViT-1B, using our proposed method on the BloodCell and CancerSlides datasets under both IID and Non-IID (Dirichlet 0.7) settings. Due to the extremely large size of ViT-1B, we were unable to evaluate its performance across all baseline methods. Therefore, the table focuses solely on the performance of our method for all ViT variants. The experimental results demonstrate that our method is effective across different sizes of ViT models, scaling well from smaller architectures like ViT-Tiny to larger ones like ViT-1B, while maintaining strong performance in both IID and Non-IID settings. Source data are provided as a Source Data file.

concentration parameter of 0.7. As summarized in Table 4, DynamicFL successfully scales across transformer architectures of varying sizes. Larger models, such as ViT-Base and ViT-1B, consistently outperform smaller models (ViT-Tiny and ViT-Small), particularly on clients with higher computational resources. For instance, on the BloodCell dataset under IID settings, ViT-1B achieves an accuracy of 94.77%, significantly outperforming ViT-Tiny, which achieves 73.11%. Even under non-IID settings, ViT-1B maintains a high accuracy of 91.26%, demonstrating its robustness in heterogeneous data distributions.

While our experiments demonstrate the scalability of DynamicFL across diverse architectures, we acknowledge that the absence of direct baseline comparisons for large-scale transformer models may introduce potential biases. Specifically, superior performance on smaller models does not necessarily guarantee the same level of improvement when scaling up to larger architectures. However, it is important to emphasize that our approach primarily focuses on lossless model transformation, enabling seamless collaboration among clients with varying computational resources. By minimizing accuracy degradation during global-to-local and local-to-global model transitions, DynamicFL ensures that knowledge transfer remains efficient and robust. Therefore, we believe that even for large-scale networks, our method can continue to deliver superior performance while maintaining fairness across heterogeneous clients.

Fairness Evaluation Among Clients of Varying Capacities

In this part, we conduct an analysis of test accuracy, i.e., the fairness among participating clients with respect to test accuracy. A fair and sustainable FL system should ensure that all nodes, regardless of their resource strength, have an equal opportunity to contribute their efforts to the collaborative learning process and share the fruits of FL equally, meaning they should achieve the same model test accuracy.To verify this point, we offer evaluation results in Fig. 2 (a), which show that DynamicFL consistently maintains identical test accuracy among strong, medium, and weak clients, while other methods show significant discrepancies in model accuracy between strong and weak clients. Taking the case of FedET as an example, the discernible discrepancy in accuracy, with a 6.31% superiority on strong clients, highlights the challenges faced by conventional approaches in mitigating the performance gap between different-sized clients. The implications of these findings are paramount in scenarios where weak clients significantly outperform their stronger counterparts, underscoring the need for methods such as DynamicFL to ensure fairness in FL applications. The equitable performance demonstrated by DynamicFL underscores its potential to address the pervasive challenges associated with model heterogeneity and provides a promising avenue for promoting fairness across diverse client profiles.

Robustness Analysis Across Varying Non-IID Scenarios

To evaluate the robustness of DynamicFL under varying degrees of data heterogeneity, we conduct experiments using different values of the Dirichlet parameter β, including 0.1, 0.3, 0.5, 0.7, and 0.9. These values represent different levels of data distribution imbalance, where smaller β values correspond to more severe Non-IID scenarios, and larger β values approximate IID distributions. The experimental results, presented in Table 5, demonstrate that DynamicFL consistently outperforms all baseline methods across all Non-IID degrees and model architectures.

DynamicFL achieves the highest accuracy in every scenario, with the performance gap being more pronounced in highly heterogeneous settings (e.g., $\beta = 0.1$ and $\beta = 0.3$). For instance, on the ResNet-18 architecture, DynamicFL surpasses the best baseline method by 1.73% under $\beta = 0.5$ and by 2.49% under $\beta = 0.1$. Similarly, on the ViT-Base architecture, DynamicFL achieves a remarkable improvement of 2.56% under $\beta = 0.5$ and 4.31% under $\beta = 0.1$. While the performance of all methods improves as the data distribution becomes closer to IID (i.e., larger β values), DynamicFL maintains a clear advantage over the baselines, demonstrating its adaptability to both IID and Non-IID scenarios.

These results highlight the robustness and generalizability of DynamicFL in FL environments with diverse levels of data heterogeneity. Its consistent performance across different β values confirms its ability to effectively address the challenges posed by Non-IID data distributions in real-world FL systems.

Scalability Evaluation with Increased Number of Clients

To evaluate the scalability of DynamicFL, we conduct experiments with 60, 120, and 180 clients on the BloodCell and CancerSlides datasets under both IID and non-IID settings. The results in Table 6 show that DynamicFL consistently outperforms baseline methods across all tested scenarios, demonstrating its robustness and scalability. As the number of clients increases, the overall accuracy of all methods decreases due to growing data heterogeneity, but DynamicFL maintains a significant performance advantage. For example, on the BloodCell dataset with 180 clients, DynamicFL achieves 72.69% accuracy with ResNet-18, outperforming the best baseline methods by 3.24%–8.91%, while on the CancerSlides dataset, it achieves 76.59% accuracy, maintaining a strong lead. DynamicFL also performs effectively across different model architectures, such as ResNet-18 and ViT-Base, with ViT-Base achieving 75.39% accuracy on the BloodCell dataset with 180 clients, significantly outperforming the 68.34% achieved by the best baseline method. These results confirm that DynamicFL scales effectively to larger numbers of clients while maintaining fairness and robustness, making it suitable for real-world FL applications.

Convergence Speed Evaluation

In Section Convergence Analysis, we present the theoretical convergence analysis of DynamicFL, indicating that DynamicFL shares the same convergence rate as the vanilla FedAvg. To further elaborate, we provide a comparison of the convergence speeds of the compared methods in Fig. 2 (b). The visualization underscores the superior convergence speed and stability exhibited by our method during the initial training phase when compared to existing approaches. This advantage arises from the intricacies inherent in model-heterogeneous FL, where knowledge transfer between two heterogeneous models occurs twice-first during aggregation and then during broadcasting. Specifically, in distillation and pruning based methods, knowledge transfer between local models of different clients has a profound impact on accuracy, leading to significant fluctuations in the accuracy of the global model during training for both approaches. In contrast, our method, thanks to the lossless knowledge transfer capabilities across heterogeneous models, demonstrates minimal accuracy fluctuations in the early stages of training. Furthermore, compared to distillation and pruning based methods, our approach exhibits swifter convergence and achieves a higher final accuracy.

Evaluation on the Ability of Enhanced Privacy Protection

In this subsection, we assess our method's performance in terms of enhanced privacy protection. Originally, FL was predicated on the belief that sharing gradients, as opposed to raw data, would not substantially endanger client privacy. Nonetheless, recent studies[46,47] have unveiled the potential for a "gradient inversion attack". In this type of attack, an adversary monitoring a client's communication with the server can start piecing together the client's confidential data. This attacker could be a malicious entity within the FL protocols. This includes a server that is honest-but-curious and aims to reconstruct the private data of its clients, or a client with similar intentions, seeking to gather private information about other clients. Most current FL methodologies, including the commonly used vanilla FedAvg, operate under a uniform model capacity assumption, utilizing identical network architectures for both local and global models. This approach, however, is susceptible to gradient inversion attacks. Our method, by contrast, shows resilience against such attacks, thanks to the combination of heterogeneous local training and homogeneous global aggregation.

We follow the methodology presented in[46], replicating their experimental setup. This involves setting the local batch size to one and uploading the updates of a single batch gradient to the central server. The results of our experiments, as depicted in Fig. 3, provide insightful observations. It can be observed that the vanilla FedAvg method offers minimal protection against privacy breaches, with private data being almost entirely reconstructible from gradients. In contrast, our approach is designed to keep the central server in the dark regarding the intricate details of local models. This strategic design effectively hinders the reconstruction of local training data, thereby significantly bolstering the privacy of edge devices. This distinctive attribute of our method showcases its strength and efficiency in safeguarding against gradient inversion attacks within the realm of FL.

Discussion

Advantages of Our Scheme

From a multi-objective optimization standpoint, the objective behind designing FL algorithms is to navigate the Pareto front within the model utility−data privacy−system efficiency space. Some studies, such as[48], posit the absence of a free lunch when addressing the three objectives in FL, suggesting that optimizing one objective may result in

https://doi.org/10.1038/s41467-025-58055-3

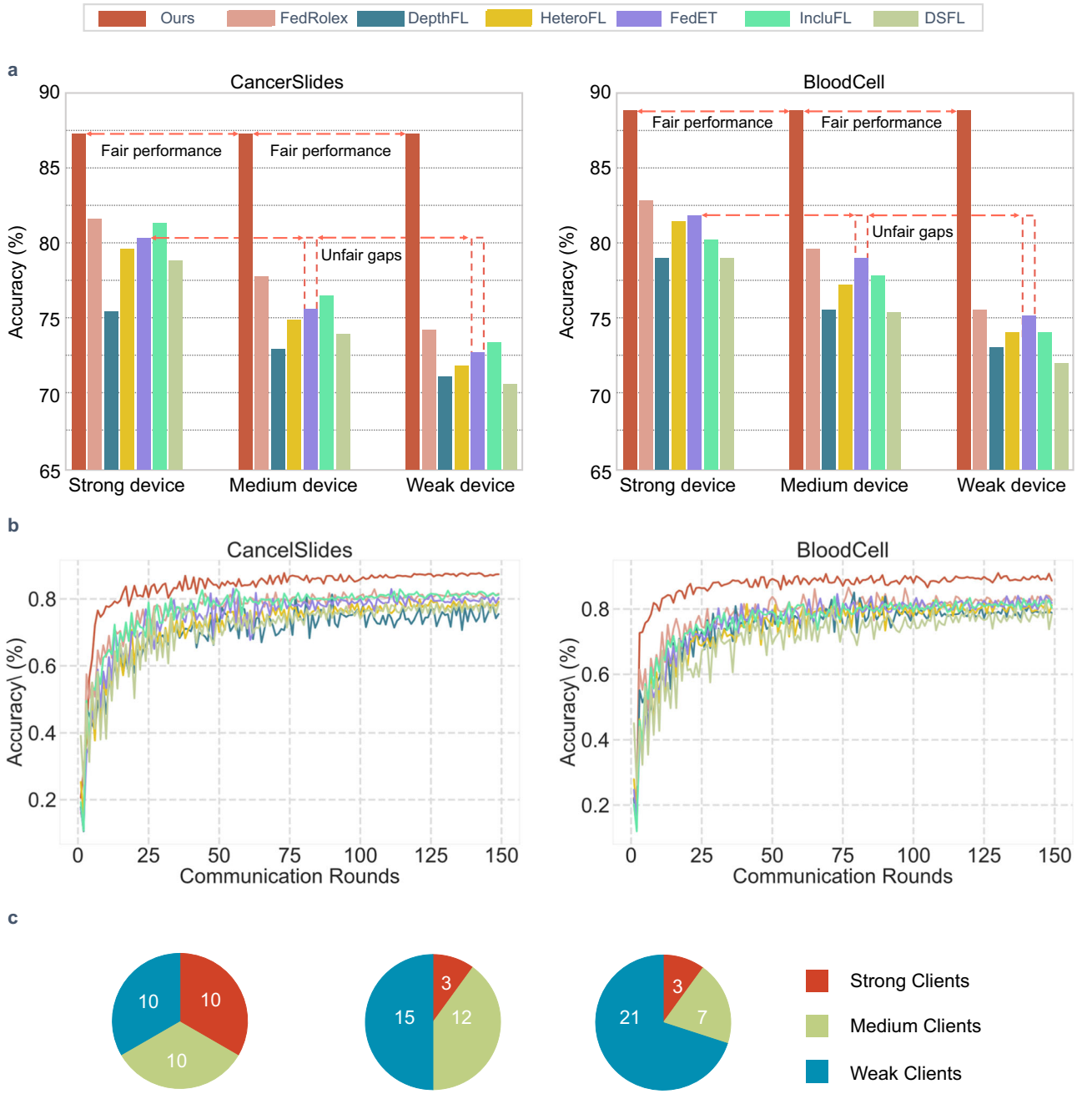

Fig. 2 | Fairness & convergence evaluation. a, Test accuracy of heterogeneous models obtained by three types of clients with different computational power on the same global test set. **b**, Convergence analysis of DynamicFL and other baselines on CancerSlides and BloodCell. **c**, Pie chart illustrating the proportions of strong, medium, and weak clients under different client distributions in the experiments. Source data are provided as a Source Data file.

a decline in others. Our method challenges this claim, which achieves fair client participation, superior model utility, enhanced data privacy, and improved system efficiency simultaneously. The detailed discussion is offered as follows:

- **Fair Clients Participation:** DynamicFL ensures fair treatment for all clients, regardless of their computational strength, actively involving them in the learning process at their full operational capability. Unlike knowledge distillation based and network pruning based approaches, DynamicFL fully transfers global model knowledge even to resource-constrained clients, without the consumption seen in other methods. In our approach, every client contributes meaningfully, fostering sustainability in the FL ecosystem.
- **Superior Model Accuracy:** Attributing to that all clients work at full capacity and knowledge is losslessly transferred across heterogeneous models, our method achieves superior test accuracy. As demonstrated by experimental results, DynamicFL achieves much better performance than the state-of-the-art knowledge distillation based and network pruning based heterogeneous FL methods.
- **Enhanced Privacy Protection:** DynamicFL enjoys a side benefit of enhanced privacy preservation. Due to the dynamic construction of local models, the server remains unaware of the actual network structures, rendering gradient inversion attacks ineffective. Experimental results (see Fig. 3) underscore the effectiveness of this model-gradients separation mechanism in mitigating data leakage risks. Notably, compared with other privacy-preserving enhancement strategies like homomorphic encryption[49] and perturbation techniques[50,51], our strategy seamlessly integrates

Table 5 | Performance evaluation of DynamicFL and other methods on ResNet and ViT-Base architectures under the same wall-clock training time

Model	Methods	Dirichlet (0.1)	Dirichlet (0.3)	Dirichlet (0.5)	Dirichlet (0.7)	Dirichlet (0.9)
ResNet-18	DSFL[26]	71.82	72.45	73.12	74.56	75.23
	FedET[27]	73.65	74.12	75.34	76.45	77.12
	IncluFL[34]	74.32	74.89	75.45	76.67	77.34
	FedMD[35]	72.89	73.23	74.12	75.01	75.56
	FedTGP[40]	73.45	74.01	75.89	76.45	78.23
	FedDF[25]	76.58	77.12	77.78	78.34	78.89
	pFedHR[39]	75.75	76.34	77.01	77.67	78.23
	FCCL[38]	74.12	76.76	77.45	78.23	79.78
	HeteroFL[28]	72.65	73.12	74.34	75.23	76.12
	DepthFL[36]	69.12	70.45	71.34	72.12	73.01
	FedRolex[37]	74.87	75.45	76.12	77.34	78.01
	DynamicFL	**78.31**	**81.48**	**82.37**	**83.14**	**84.58**
ViT-Base	DSFL[26]	70.54	71.23	72.01	73.12	74.23
	FedET[27]	73.12	73.89	74.67	75.34	76.45
	IncluFL[34]	74.01	74.45	75.12	75.89	76.67
	FedMD[35]	72.65	73.01	73.67	74.34	75.12
	FedDF[25]	75.23	75.78	76.34	76.89	77.45
	DynamicFL	**79.54**	**84.30**	**84.68**	**85.09**	**86.90**

The table presents the performance of DynamicFL and baseline methods on ResNet-18 and ViT-Base architectures under different levels of Non-IID data heterogeneity. Non-IID distributions are introduced using the Dirichlet distribution Dir(β), with concentration parameters β ranging from 0.1 to 0.9. Results highlighted in bold indicate superior performance. The results demonstrate that DynamicFL consistently achieves better performance across varying levels of data heterogeneity, showcasing its robustness and adaptability. Source data are provided as a Source Data file. Source data are provided as a Source Data file.

Table 6 | Performance evaluation of DynamicFL and other methods with more clients

		CancerSlides			BloodCell		
Model	Methods	60	120	180	60	120	180
ResNet-18	DSFL[26]	71.23	69.34	67.12	69.56	68.12	66.45
	FedET[27]	73.12	71.45	69.34	71.23	70.01	67.89
	IncluFL[34]	72.45	70.78	68.56	70.89	69.56	67.23
	FedMD[35]	70.34	68.56	66.89	68.45	67.34	65.78
	FedTGP[40]	74.12	72.34	72.78	70.78	67.12	64.89
	FedDF[25]	74.89	72.67	70.56	72.34	71.12	69.45
	pFedHR[39]	73.34	71.78	69.89	71.89	70.45	68.56
	FCCL[38]	72.78	70.89	68.45	70.34	69.01	67.12
	HeteroFL[28]	70.01	68.34	66.12	66.78	64.45	62.78
	DepthFL[36]	68.45	66.89	65.12	66.12	65.01	63.78
	FedRolex[37]	73.45	71.78	69.56	71.23	69.89	67.78
	DynamicFL	**82.61**	**80.15**	**76.59**	**80.88**	**76.31**	**72.69**
ViT-Base	DSFL[26]	70.89	68.78	66.34	69.23	67.78	65.56
	FedET[27]	72.45	70.56	68.12	71.12	69.45	67.01
	IncluFL[34]	71.78	69.89	67.45	70.34	68.78	66.45
	FedMD[35]	69.89	67.78	66.12	68.45	66.89	65.01
	FedDF[25]	73.89	71.56	69.12	72.34	70.78	68.34
	DynamicFL	**80.24**	**78.26**	**74.72**	**84.36**	**78.11**	**75.39**

The table presents the performance of DynamicFL and other baseline methods under varying numbers of clients (60, 120, and 180). Experiments were conducted on two datasets, CancerSlides and BloodCell, using ResNet-18 and ViT-Base as network architectures. Results highlighted in bold indicate superior performance. The results demonstrate that DynamicFL scales effectively to larger client populations, highlighting its robustness and adaptability. Source data are provided as a Source Data file.

into the entire learning process without introducing additional computational burdens or performance degradation.

- **Improved System Efficiency:** In DynamicFL, the process of structural re-parameterization introduces additional computation costs as a trade-off, seemingly impacting system efficiency. However, the introduction of a lightweight global model mitigates this effect by reducing communication overhead. The experimental results (see Table 3) indicate that, within the same training time budget, our lightweight model outperforms state-of-the-art methods employing larger global models. Moreover, the theoretical convergence analysis provided in Convergence Analysis shows that DynamicFL achieves the same convergence performance with FedAvg, and empirical results (see Fig. 2 (b)) suggest that our method achieves swifter convergence compared with the state-of-the-art baselines.

Future Challenges and Outlook

While DynamicFL effectively enhances fairness in federated learning, real-world deployment presents additional challenges. One key issue is communication synchronization among participating healthcare institutions. Existing research has yet to explore asynchronous model-heterogeneous federated learning, meaning that all participating institutions must maintain synchronized communication during training. Additionally, frequent transmission of model parameters in federated learning imposes significant communication overhead, which could further strain network resources. Future work could explore adaptive scheduling mechanisms to mitigate communication overhead, ensuring more efficient federated learning in real-world deployments.

Methods

Ethical Statement

Our study utilizes publicly available datasets to conduct federated learning for collaborative training across clients with varying computational resources. As no human participants, personal data, or ethical concerns are involved, this research does not require ethical approval.

Overall Workflow

In this subsection, we elaborate the methodology of DynamicFL. We consider FL across N heterogeneous edge clients with diverse computational capabilities. Each client i can only access its own private dataset $\mathcal{D}_i = \{(\mathbf{x}_j^i, y_j^i)\}$ where $\mathbf{x}$ and y denote the input features and corresponding class labels, respectively. The overall algorithmic workflow is summarized in 1 In the following, we elaborate the main steps of DynamicFL:

Initialization. Define the global model as the convolutional neural network, which is still widely used in real-world applications due to the

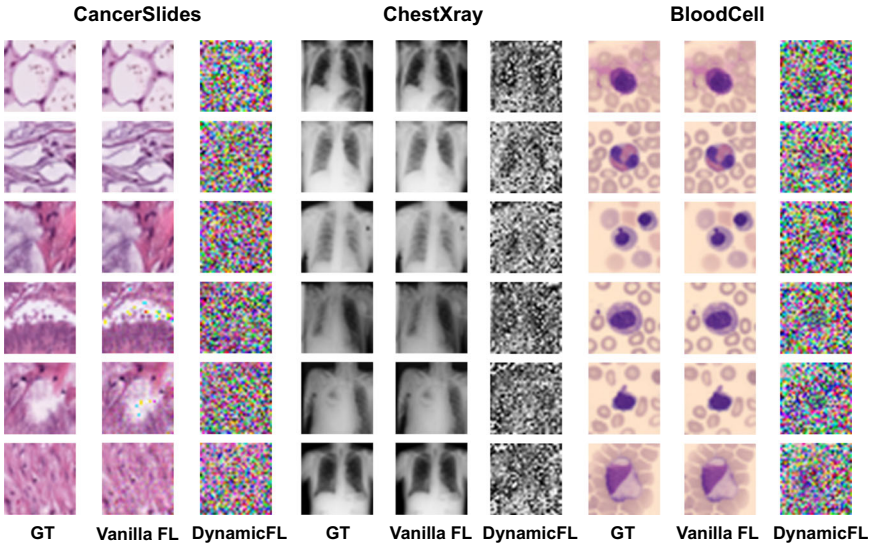

Fig. 3 | Privacy evaluation. The existing model-heterogeneous FL methods and Vanilla FedAvg are deficient in their defense against gradient inversion attacks. This shortcoming exposes a significant vulnerability: a malicious central server can easily exploit the received gradient information to reconstruct clients' private data, thereby posing a substantial privacy risk. These methods resort to implementing established encryption techniques within FL, but this approach often leads to reduced system efficiency and accuracy loss. DynamicFL, on the other hand, offers a more effective and robust solution. In the DynamicFL framework, the central server remains unaware of the specific private models employed by individual clients. This lack of information severely restricts the server's capability to reconstruct semantic information, significantly reducing the potential for privacy breaches and thus bolstering the protection of client data. Source data are provided as a Source Data file.

advantage of parallel computing and lightweight memory footprint. The server initializes the model parameters and hyper-parameters and distributes the global model to all clients. In the first round, once the global model is received, the client i expands the $K \times K$ convolution layers to diverse branches adopted in DBB[32] from bottom to top according to their computational resources:

$$\zeta_i^{(0)} \leftarrow \mathrm{REP}(\omega_g^{(0)}). \tag{1}$$

The stronger the client, the more layers it can expand. Upon the derived local model $\zeta_i^{(0)}$, we perform local training for E epochs:

$$\begin{aligned} \zeta_i^{(0)} &\leftarrow \zeta_i^{(0)} - \eta \nabla \ell(\zeta_i^{(0)}; \mathcal{D}_i), \\ S_i^l &\leftarrow \nabla \ell(\zeta_i^{(0)}; \mathcal{D}_i). \end{aligned} \tag{2}$$

We record the gradient of the last epoch as S_i^l, which reflects the sensitivity of each branch of the local model to local knowledge. The local model is then transformed back to the original plain structure and uploaded to the server. Since all uploaded local models have the same structure as the global model, we can perform the aggregation operation to derive the updated global model $\omega_g^{(0)}$.

Heterogeneous Local Training. The new global model is distributed to local clients, upon which the next round of local training is conducted. Considering local clients have diverse and limited computational resources, we dynamically adjust computational workloads by selecting operations that significantly contribute to performance, instead of re-parameterizing all candidate operations as done in DBB. The resource-adaptive models modulation is tailored according the local and global gradient information of FL. Specifically, in the t-th round, relying on the new global model $\omega_g^{(t-1)}$ and the stale local model $\zeta_i^{(t-1)}$ in the last round, we conduct structural re-parameterization to derive a temporary local model $\epsilon_i^{(t)}$ that has the same output as $\omega_g^{(t-1)}$ while with the same structure as $\zeta_i^{(t-1)}$:

$$\epsilon_i^{(t)} \leftarrow \mathrm{REP}(\omega_g^{(t-1)}, \zeta_i^{(t-1)}), \tag{3}$$

where REP(·) represents the re-parameterization operation. We perform local training on $\epsilon_i^{(t)}$ along with the private dataset $\mathcal{D}_i$ for only one epoch, and obtain the gradient information S_i^g:

$$\begin{aligned} \epsilon_i^{(t)} &\leftarrow \epsilon_i^{(t)} - \eta \nabla \ell(\epsilon_i^{(t)}; \mathcal{D}_i), \\ S_i^g &\leftarrow \nabla \ell(\epsilon_i^{(t)}; \mathcal{D}_i), \end{aligned} \tag{4}$$

where S_i^g reflects the sensitivity of each branch of the local model to the global knowledge, *i.e.*, the aggregation result of all participants.

S_i^g and S_i^l offer useful cues to reflect the contribution of branches to the global aggregation. According to them, we dynamically evolve the network structures of the local model to remove redundant operations with little contribution while further expanding important operations with significant contribution, and obtain the new local model $\zeta_i^{(t)}$:

$$\zeta_i^{(t)} \leftarrow \mathrm{DYMM}(\epsilon_i^{(t)}; S_i^g, S_i^l). \tag{5}$$

The details of DYMM(·) for how to use S_i^g, S_i^l to perform dynamic model modulation can be found in Section Dynamic Model Modulation.

Client i then performs local training along with its private data on the new local model $\zeta_i^{(t)}$ with full operational capability:

$$\zeta_i^{(t)} \leftarrow \zeta_i^{(t)} - \eta \nabla \ell(\zeta_i^{(t)}; \mathcal{D}_i). \tag{6}$$

where the number of epochs executed is $E - 1$, since one epoch has already been done in Eq. (4). The gradient information of the last epoch is recorded as the updated S_i^l:

$$S_i^l \leftarrow \nabla \ell(\zeta_i^{(t)}; \mathcal{D}_i). \tag{7}$$

It is worth mentioning a possible scenario: in this round, client i may have a smaller computational budget than in the last round due to other active routines, which makes it unable to afford the local model from the previous round. In this case, the steps correspond to Eq. (3), (4) and (5) cannot be performed. Instead, we conduct

re-parameterization based on S_i^l only to derive the new model:

$$\zeta_i^{(t)} \leftarrow \mathrm{REP}(\omega_g^{(t-1)}, S_i^l). \tag{8}$$

Then, local training on $\zeta_i^{(t)}$ is performed for E epochs, as shown in Eq. (6).

Homogeneous Global Aggregation. Once Client i completes the local training, the local model $\zeta_i^{(t)}$ is transformed back to the original global model structure by re-parameterization:

$$\omega_i^{(t)} \leftarrow \mathrm{REP}(\zeta_i^{(t)}, \omega_g^{(t-1)}), \tag{9}$$

$\omega_i^{(t)}$ has the same output as the local model $\zeta_i^{(t)}$ while with the same structure as the global model $\omega_g^{(t-1)}$. After the central server receives the uploaded $\{\omega_i^{(t)}\}_{i=1}^N$ from all clients, the global aggregation can be done since $\{\omega_i^{(t)}\}_{i=1}^N$ share the same network structure:

$$\omega_g^{(t)} \leftarrow \frac{1}{N}\sum_{\omega_i}^{(t)}. \tag{10}$$

The equivalent transformations of operations during re-parameterization ensure lossless knowledge transfer across heterogeneous local models. The central server then distributes the new global model $\omega_g^{(t)}$ to local clients to start the next round of local training.

Algorithm 1. Dynamic Federated Learning

Input: Initialized global model, N edge devices with private datasets $\{\mathcal{D}_i\}_{i=1}^N$, communication round number T, learning rate η, epoch number E.
Output: The final global model ω_g^T.
for each communication round $t = 1, \ldots, T$ **do**
  Send the global model $\omega_g^{(t-1)}$ to the edge devices
  for each device $i = 1, 2, \ldots, N$ **in parallel do**
    Receive the global model $\omega_g^{(t-1)}$
    $\zeta_i^{(t)} \leftarrow$ **HeterogeneousLocalTraining** $(i, \omega_g^{(t-1)})$
    Re-parameterization to global model: $\omega_i^{(t)} \leftarrow \mathrm{REP}(\zeta_i^{(t)}, \omega_g^t)$
    Return $\omega_i^{(t)}$ to server
  $\omega_g^{(t)} \leftarrow \frac{1}{N}\sum \omega_i^{(t)}$
return ω_g^T
HeterogeneousLocalTraining $(i, \omega_g^{(t-1)})$**:**
  Re-parameterization to local model: $\epsilon_i^{(t)} \leftarrow \mathrm{REP}(\omega_g^{(t-1)}, \zeta_i^{(t-1)})$
  Train one local epoch: $\epsilon_i^{(t)} \leftarrow \epsilon_i^{(t)} - \eta\nabla\ell(\epsilon_i^{(t)}; \mathcal{D}_i)$
  Record gradient: $S_i^g \leftarrow \nabla\ell(\epsilon_i^{(t)}; \mathcal{D}_i)$
  Re-parameterization according to score S_i^g, S_i^l: $\zeta_i^{(t)} \leftarrow \mathrm{DYMM}(\epsilon_i^t; S_i^g, S_i^l)$
  for local epoch $k = 1, 2, ..., E-1$ **do**
    $\zeta_i^{(t)} \leftarrow \zeta_i^{(t)} - \eta\nabla\ell(\zeta_i^{(t+1)}; \mathcal{D}_i)$
    if $k = E-1$ **then**
      Record gradient: $S_i^l \leftarrow \nabla\ell(\zeta_i^{(t)}; \mathcal{D}_i)$
  return $\zeta_i^{(t)}$

Dynamic Model Modulation

In this subsection, we present the details of the DYMM(·) operator to perform dynamic model modulation based on the local knowledge from the local private dataset and the global knowledge from the global aggregation.

We dynamically identify the important operations and redundant operations of local models according to the gradient information. Some works in the literature[52,53] have shown that the gradient of each weight of the model in the training process can effectively reflect the sensitivity of the weight to data. Inspired by works[54,55], we use the following metric to measure the salience of each weight:

$$\mathcal{S}(\theta_i) = \frac{\partial\ell}{\partial\theta}(\theta_i), \tag{11}$$

where $\theta_i \in \boldsymbol{\theta}$ is the parameter of the model.

Local knowledge in FL is easy to obtain during local training. In the $(t-1)$-th round, we record the last local epoch gradient information by Eq. (7) and (11):

$$S_i^l(\boldsymbol{\theta}^k) = \sum_j^n \frac{\partial\ell}{\partial\theta_j^k} \odot \theta_j^k, \tag{12}$$

where $\boldsymbol{\theta}^k$ is the k-th branch in the local model $\zeta_i^{(t-1)}$, and θ_j^k is the j-th parameter of the branch k. $S_i^l(\boldsymbol{\theta}^k)$ is leveraged for dynamic model modulation in the next round.

The global knowledge in FL is implicitly encoded into the global model, which is the aggregation result of all participating clients. To extract global information, according to the received global model $\omega_g^{(t-1)}$ in the t-th round and the last round local model $\zeta_i^{(t-1)}$, re-parameterization is conducted to get the temporary model ϵ_i^t through Eq. (3). Here ϵ_i^t is a re-parameterized model of $\omega_g^{(t-1)}$, which has the same structure as $\zeta_i^{(t-1)}$. We use ϵ_i^t to perform one-epoch local training with the private dataset, from which the derived gradient information reflects the sensitivity of the branches in the local model $\zeta_i^{(t-1)}$ to the global information:

$$S_i^g(\psi^k) = \sum_j^n \frac{\partial\ell}{\partial\psi_j^k} \odot \psi_j^k, \tag{13}$$

where ψ^k is the k-th branch in the local model $\epsilon_i^{(t)}$, and ψ_j^k is the j-th parameter of branch k.

We regard branches with small S_i^l and S_i^g as redundant ones; the branches with a small S_i^l but large S_i^g as important ones since they are sensitive to global knowledge; the rest branches as common ones. In Eq. (5), we first merge redundant branches into common branches to reduce the size of the local model:

$$\psi^{(\text{common}')} \leftarrow \psi^{(\text{common})} + \psi^{(\text{redundant})}, \tag{14}$$

where $\psi^{(\text{common}')}$ is the new common branch after merging; $\psi^{(\text{common})}$ and $\psi^{(\text{redundant})}$ represent common and redundant branches, respectively. Then we expand the important branches to adapt the client's computing capabilities:

$$\psi^{(\text{important}')} \leftarrow \psi^{(\text{important})} - (\psi^{(1)} + \cdots + \psi^{(n)}), \tag{15}$$

where $\psi^{(\text{important}')}$ are the expanded parameters of the important branch. In order to ensure the same output, we need to subtract the parameters of the new branch $\{\psi^{(i)}\}_{i=1}^n$ from the parameters of the original important branch $\psi^{(\text{important})}$. The parameters of the new branch are generated randomly.

To ensure the stability and effectiveness of training, we adopt a fixed local reparameterization strategy for transformer-based models throughout the training process, unless computational resources change. Transformer architectures are particularly sensitive to parameter variations, and frequent changes to the reparameterization strategy may adversely affect their convergence and performance.

Lossless Knowledge Transfer

This section introduces the re-parameterization techniques for CNNs and transformers, demonstrating how these methods enable flexible structural transformations while preserving model outputs.

Re-parameterization for CNN. As indicated by works[55,56], the 2D convolutions hold the property of additivity:

$$\boldsymbol{I} \circledast \boldsymbol{F}^{(1)} + \boldsymbol{I} \circledast \boldsymbol{F}^{(2)} = \boldsymbol{I} \circledast \left(\boldsymbol{F}^{(1)} + \boldsymbol{F}^{(2)}\right). \tag{16}$$

where $\boldsymbol{I}$, $\boldsymbol{F}^{(1)}$ and $\boldsymbol{F}^{(2)}$ are the input and kernels, respectively. The above equation is satisfied even with different kernel sizes. Some widely used

operations in CNN —- average pooling and batch normalization —- can be converted into a convolution operation. The above additivity property ensures that a single convolution can be equivalently transformed to multi-branch operations, and vice versa. The equivalent transformations of operations guarantee lossless knowledge transfer, since the model outputs are not changed along with the network structure adjustment.

Re-parameterization for Transformer. The re-parameterization technique can also be adapted to transformer architectures. In the case of transformers, the linear layers hold the property of additivity, similar to the convolution operation in CNNs:

$$\boldsymbol{X}\cdot\boldsymbol{W}^{(1)}+\boldsymbol{X}\cdot\boldsymbol{W}^{(2)}=\boldsymbol{X}\cdot\left(\boldsymbol{W}^{(1)}+\boldsymbol{W}^{(2)}\right), \tag{17}$$

where $\boldsymbol{X}$ is the input feature matrix, and $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ are the weight matrices of two linear layers. This equation implies that multiple parallel linear layers can be equivalently merged into a single linear layer, or vice versa.

Moreover, operations commonly used in transformer architectures, such as layer normalization and residual connections, can also be transformed into equivalent forms compatible with the re-parameterization technique. Specifically, layer normalization can be expressed as an affine transformation, which can be absorbed into the parameters of linear layers. Residual connections, being additive in nature, align naturally with the additivity property of linear operations.

The re-parameterization process for transformers involves expanding a single linear layer into multiple parallel linear layers, optionally combined with normalization layers (e.g., batch normalization or layer normalization). These parallel branches can then be merged back into a single equivalent linear layer without loss of information. This guarantees lossless knowledge transfer, as the outputs of the model remain unchanged before and after the structural reconfiguration:

$$\boldsymbol{X}\cdot\boldsymbol{W}\ \text{merged}=\boldsymbol{X}\cdot(\boldsymbol{W}^{(1)}+\boldsymbol{W}^{(2)}), \tag{18}$$

where $\boldsymbol{W}$merged is the weight matrix of the merged linear layer.

It is worth noting that, according to Eq. (16) and (17), the re-parameterization process in DynamicFL ensures mathematical equivalence between the original and transformed model structures, guaranteeing that the input-output mapping remains unchanged. As a result, when evaluating the model on the same test set, its accuracy remains identical, thereby ensuring fairness in the models obtained by institutions with different computational resources.

Convergence Analysis

In this subsection, we present the convergence analysis of DynamicFL. We consider the following standard assumptions commonly made in FL analysis[57–59]:

Assumption 1. (L-smoothness and σ-uniformly bounded gradient variance).

(a) F is L-smooth, i.e., $F(u)\le F(x)+\langle\nabla F(x),u-x\rangle+\frac{1}{2}L\|u-x\|^2$ for any $u,x\in\mathbb{R}^d$.

(b) There exists a constant $G_{max}>0$ such that: $\mathbb{E}[||\nabla F^{(i)}(x)||^2]\le G_{max}^2,\quad \forall i\in[N],\forall\mathbf{x}\in\mathbb{R}^d$, where $\nabla F^{(i)}(x)$ is an unbiased stochastic gradient of $f^{(i)}$ at x.

(c) $\nabla f(x)$ has σ^2 -bounded variance, i.e., $\mathbb{E}_{\xi\sim\mathcal{S}_i}\|\nabla F_i(\mathbf{x})-\nabla f_i(\mathbf{x})\|\le\sigma^2,\quad\forall i\in[N],\forall\mathbf{x}\in\mathbb{R}^d$.

Lemma 2. For any reparameterization of a convolutional layer l that can be represented as a summation of N convolutional branches with weights $W_{l,n}^{(t)}$ and binary receptive field mask $M_{l,n}$, for n = 1, ..., N, its gradient descent update can be respresented as:

$$W_l^{(t+1)}\Leftarrow W_l^{(t)}-\lambda^{(t)}f\left(\mathcal{G}_l\odot\frac{\partial\mathcal{L}}{\partial W_l^{(t)}},W_l^{(t)},\ldots,\mathcal{G}_l\odot\frac{\partial\mathcal{L}}{\partial W_l^{(0)}},W_l^{(0)}\right), \tag{19}$$

where $\mathcal{G}_l=\sum_{n=1}^{N}M_{l,n}$. Therefore, it can be seen as spatial gradient scaling applied to the original convolution. Here we assume $\mathcal{G}_l\le\mathcal{G}$. In FL, it can be expressed that for the local model $\zeta_{i,k}^{(t)}$ on edge client i,:

$$\zeta_{i,k+1}^{(t)}\leftarrow\zeta_{i,k}^{(t)}-\eta\nabla\ell(\zeta_i^{(t+1)};\mathcal{D}_i). \tag{20}$$

Here, we note that the variable k denotes the current local epoch, and each client's structure of $\zeta_{i,k}^{(t)}$ may differ from one another. We can reparameterize $\zeta_{i,k}^{(t)}$ as $\phi_i^{(t)}$ with the same structure as the global model. Specifically, we have:

$$\phi_{i,k+1}^{(t)}\leftarrow\phi_{i,k}^{(t)}-\mathcal{G}_i^{(t)}\odot\eta\nabla\ell(\phi_i^{(t+1)};\mathcal{D}_i). \tag{21}$$

This reparameterization ensures that the structure of $\phi^{(t)}i$ is consistent across all clients, matching that of the global model. As a result, we can aggregate the virtual sequences on each client to obtain the global model $\omega^{(g)}t$, which satisfies:

$$\mathbb{E}\left[\left\|\phi_{i,k}^{(t)}-\omega_t^{(g)}\right\|^2\right]\le4\eta^2K^2\mathcal{G}^2G^2\quad\forall t,k,i. \tag{22}$$

Theorem 3. The sequence generated by our method with stepsize $\eta\le1L$ satisfies

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f\left(w_t^{(g)}\right)\right\|^2\right]\le\frac{2}{\eta T}\left(\mathbb{E}\left[f\left(w_1^{(g)}\right)\right]-f\left(w_T^{(g)}\right)\right)+4\eta^2\mathcal{G}^2L^2G^2K^2+\frac{L}{N}\eta\sigma^2. \tag{23}$$

Corollary 4. When the function f is lower bounded with $f(w_1^{(g)})-f^*\le\Delta$ and the number rounds T is large enough, then set the stepsize $\eta=\frac{\sqrt{N}}{L\sqrt{T}}$ yields

$$\frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f\left(w_t^{(g)}\right)\right\|^2\right]=O\left(\frac{2L\Delta+\sigma^2}{\sqrt{NT}}+\frac{N}{T}\right). \tag{24}$$

The dominance of the first term in our algorithm ensures that it shares the same convergence speed, $O(1/\sqrt{NT})$, as the vanilla FedAvg.

Statistics & Reproducibility

This study is based on a publicly available dataset. No statistical method was used to predetermine sample size. No data were excluded from the analyses. Since the dataset is pre-collected and publicly available, no randomization or blinding was applicable. The machine learning models were trained using standard procedures, and all experiments were conducted with fixed hyperparameters unless otherwise specified.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets used in this study are publicly available. The CancerSlides dataset and BloodCell dataset are available at Zenodo (https://doi.org/10.5281/zenodo.10519652) under the CC *BY-NC* license. The ChestXray dataset is accessible through the COVID-19 Radiography Database on Kaggle (https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data). The CIFAR-10 and CIFAR-100 datasets

can be obtained from the University of Toronto's website (https://www.cs.toronto.edu/~kriz/cifar.html). Source data are provided with this paper.

Code availability

The complete source code used in this study is publicly available in the DynamicFL repository (https://github.com/paridis-11/DynamicFL) under the Apache-2.0 license, and can be cited as[60].

References

1. Kaissis, G. et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nat. Mach. Intell.* **3**, 473–484 (2021).
2. Mhasawade, V., Zhao, Y. & Chunara, R. Machine learning and algorithmic fairness in public and population health. *Nat. Mach. Intell.* **3**, 659–666 (2021).
3. Seyyed-Kalantari, L. et al. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nat. Med.* **27**, 2176–2182 (2021).
4. World Health Organization, *Ethics and governance of artificial intelligence for health: WHO guidance*. WHO, (2021).
5. Torabi, F. et al. A common framework for health data governance standards. *Nat. Med.* **30**, 26–29 (2024).
6. Price, W. & Cohen, I. Privacy in the age of medical big data. *Nat. Med.* **25**, 37–43 (2019).
7. Obermeyer, Z., Powers, B., Vogeli, C. & Mullainathan, S. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* **366**, 447–453 (2019).
8. Chen, R. J. et al. Algorithmic fairness in artificial intelligence for medicine and healthcare. *Nat. Biomed. Eng.* **7**, 719–742 (2023).
9. Vyas, D. A., Eisenstein, L. G. & Jones, D. S. Hidden in plain sight - reconsidering the use of race correction in clinical algorithms. *N. Engl. J. Med.* **383**, 874–882 (2020).
10. Crear-Perry, J., Maybank, A., Keeys, M., Mitchell, N. & Godbolt, D. Moving towards anti-racist praxis in medicine. *Lancet* **396**, 451–453 (2020).
11. 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*. PMLR, pp. 1273–1282. (2017).
12. Kaissis, G. A. et al. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2**, 305–311 (2020).
13. Bai, X. et al. Advancing covid-19 diagnosis with privacy-preserving collaboration in artificial intelligence. *Nat. Mach. Intell.* **3**, 1081–1089 (2021).
14. Karargyris, A. et al. Federated benchmarking of medical artificial intelligence with medperf. *Nat. Mach. Intell.* **5**, 799–810 (2023).
15. Zhang, F. et al. "No one idles: Efficient heterogeneous federated learning with parallel edge and server computation," in *International Conference on Machine Learning*. PMLR, pp. 41 399–41 413 (2023).
16. ADPPA, "American data privacy and protection act," [Online]. Available: https://www.congress.gov/bill/117th-congress/house-bill/8152 (2022).
17. GDPR, "General data protection regulation," [Online]. Available: https://gdprinfo.eu/ (2016).
18. Yang, J., Soltan, A. & Eyre, D. E. A. Algorithmic fairness and bias mitigation for clinical machine learning with deep reinforcement learning. *Nat. Mach. Intell.* **5**, 884–894 (2023).
19. Lin, M. et al. Improving model fairness in image-based computer-aided diagnosis. *Nat. Commun.* **14**, 6261 (2023).
20. Lin, S. et al. Overhead-free noise-tolerant federated learning: A new baseline. *Mach. Intell. Res.* **21**, 526–537 (2024).
21. Zhou, Z. et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **107**, 1738–1762 (2019).
22. Wu, C., Wu, F., Lyu, L., Huang, Y. & Xie, X. Communication-efficient federated learning via knowledge distillation. *Nat. Commun.* **13**, 2032 (2022).
23. Huang, Y., Gupta, S., Song, Z., Li, K. & Arora, S. "Evaluating gradient inversion attacks and defenses in federated learning," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241 (2021).
24. He, C., Annavaram, M. & Avestimehr, S. "Group knowledge transfer: Federated learning of large cnns at the edge," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 14068–14080 (2020).
25. Lin, T., Kong, L., Stich, S.U. & Jaggi, M. "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363 (2020).
26. Itahara, S., Nishio, T., Koda, Y., Morikura, M. & Yamamoto, K. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Trans. Mob. Comput.* **22**, 191–205 (2021).
27. Cho, Y.J., Manoel, A., Joshi, G., Sim, R. & Dimitriadis, D. "Heterogeneous ensemble knowledge transfer for training large models in federated learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, 7 pp. 2881–2887 (2022).
28. Diao, E., Ding, J. & Tarokh, V. "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *International Conference on Learning Representations*, (2021).
29. Horvath, S. et al. "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 12 876–12 889.
30. Mei, Y., Guo, P., Zhou, M. & Patel, V. "Resource-adaptive federated learning with all-in-one neural composition," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 4270–4284.
31. Huang, T., You, S., Wang, F., Qian, C. & Xu, C. "Knowledge distillation from a stronger teacher," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 716–33 727 (2022).
32. Ding, X., Zhang, X., Han, J. & Ding, G. "Diverse branch block: Building a convolution as an inception-like unit," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10 886–10 895 (2021).
33. Ding, X. et al. "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13 733–13 742 (2021).
34. Liu, R. et al. "No one left behind: Inclusive federated learning over heterogeneous devices," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, p. 3398-3406 (2022).
35. Li, D. & Wang, J. Fedmd: Heterogenous federated learning via model distillation. In *Advances in Neural Information Processing Systems Workshop*, (2019).
36. Kim, M., Yu, S., Kim, S. & Moon, S.-M. "Depthfl: Depthwise federated learning for heterogeneous clients," in *International Conference on Learning Representations*, (2022).
37. Alam, S., Liu, L., Yan, M. & Zhang, M. "Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 677–29 690 (2022).
38. Huang, W., Ye, M., Shi, Z. & Du, B. Generalizable heterogeneous federated cross-correlation and instance similarity learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 712–728 (2024).
39. Wang, J. et al. "Towards personalized federated learning via heterogeneous model reassembly," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 29 515–29 531 (2024).

40. Zhang, J., Liu, Y., Hua, Y. & Cao, J. "Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, pp. 16 768–16 776 (2024).
41. Kather, J. N., Halama, N. & Marx, A. "100,000 histological images of human colorectal cancer and healthy tissue," *Zenodo*, (2018).
42. Kather, J. N. et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Med.* **16**, 1–22 (2019).
43. Rahman, T., Chowdhury, M. and Khandakar, A. "Covid-19 radiography database," *Kaggle: San Francisco, CA, USA*, (2020).
44. Chowdhury, M. E. H. et al. Can ai help in screening viral and covid-19 pneumonia?". *IEEE Access* **8**, 132665–132676 (2020).
45. Acevedo, A. et al. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data brief*. **30**, 105474 (2020).
46. Zhu, L., Liu, Z. & Han, S. "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, vol. 32, (2019).
47. Geiping, J., Bauermeister, H., Dröge, H. & Moeller, M. "Inverting gradients-how easy is it to break privacy in federated learning?" in *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947 (2020).
48. Zhang, X., Gu, H., Fan, L., Chen, K. & Yang, Q. No free lunch theorem for security and utility in federated learning. *ACM Trans. Intell. Syst. Technol.* **14**, 1–35 (2022).
49. Zhang, C. et al. "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *2020 USENIX annual technical conference (USENIX ATC 20)*, pp. 493–506 (2020).
50. Kairouz, P., Liu, Z. and Steinke, T. "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, pp. 5201–5212 (2021).
51. Agarwal, N., Kairouz, P. & Liu, Z. The skellam mechanism for differentially private federated learning. *Adv. Neural Inf. Process. Syst.* **34**, 5052–5064 (2021).
52. Lee, N., Ajanthan, T. & Torr, P. H. "Snip: Single-shot network pruning based on connection sensitivity," in *International Conference on Learning Representations*, (2018).
53. Wang, C., Zhang, G. & Grosse, R. "Picking winning tickets before training by preserving gradient flow," in *International Conference on Learning Representations*, (2020).
54. Tanaka, H., Kunin, D., Yamins, D. L. & Ganguli, S. Pruning neural networks without any data by iteratively conserving synaptic flow. *Adv. Neural Inf. Process. Syst.* **33**, 6377–6389 (2020).
55. Huang, T. et al. "Dyrep: Bootstrapping training with dynamic re-parameterization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 588–597 (2022).
56. Ding, X., Guo, Y., Ding, G. & Han, J. "Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," in *Proceedings of the IEEE/CVF Int Confer. Comput. Vision (ICCV)*, **10**, 1911–1920 (2019).
57. Stich, S. U. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, (2019).
58. Avdiukhin, D. & Kasiviswanathan, S. "Federated learning under arbitrary communication patterns," in *International Conference on Machine Learning*. PMLR, pp. 425–435 (2021).
59. Nguyen, J. et al. "Federated learning with buffered asynchronous aggregation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3581–3607 (2022).
60. Zhang, F. "Towards fairness-aware and privacy-preserving enhanced collaborative learning for healthcare," https://doi.org/10.5281/zenodo.14942270.

Acknowledgements

This work was supported in part by National Key Research and Development Program of China under Grant 2023YFC2509100 (X.L.), an in part by National Natural Science Foundation of China under Grant 92270116 (X.L.).

Author contributions

X.L. and X.J. coordinated and supervised the research project. F.Z. proposed the idea for the DynamicFL framework, implemented the models, conducted the experiments, and wrote the manuscript. D.Z., G.B., J.J., and Q.Y. discussed and analyzed the results. X.L., X.J., and Q.Y. provided feedback and contributed to refining the manuscript. All authors contributed to discussion of the DynamicFL framework.

Competing interests

The authors declare the following competing interests: The encryption algorithm presented in this paper is subject to a patent application. The patent applicant is Harbin Institute of Technology, and the inventors are Xianming Liu, Feilong Zhang, Shiyi Lin, Deming Zhai, Junjun Jiang, and Xiangyang Ji. The application number is CN202411336754.4, and its current status is initiative for examination as to substance. The other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-58055-3.

Correspondence and requests for materials should be addressed to Xiangyang Ji or Xianming Liu.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.