# PracMHBench: Re-evaluating Model-Heterogeneous Federated Learning Based on Practical Edge Device Constraints

Yuanchun Guo, Bingyan Liu*, Yulong Sha, Zhensheng Xian
{gyc2001, bingyanliu, shayulong, xsin}@bupt.edu.cn
*School of Computer Science , Beijing University of Posts and Telecommunications*, Beijing, China

arXiv:2509.08750v1 [cs.LG] 4 Sep 2025

***Abstract*—Federating heterogeneous models on edge devices with diverse resource constraints has been a notable trend in recent years. Compared to traditional federated learning (FL) that assumes an identical model architecture to cooperate, model-heterogeneous FL is more practical and flexible since the model can be customized to satisfy the deployment requirement. Unfortunately, no prior work ever dives into the existing model-heterogeneous FL algorithms under the practical edge device constraints and provides quantitative analysis on various data scenarios and metrics, which motivates us to rethink and re-evaluate this paradigm. In our work, we construct the first system platform PracMHBench to evaluate model-heterogeneous FL on practical constraints of edge devices, where diverse model heterogeneity algorithms are classified and tested on multiple data tasks and metrics. Based on the platform, we perform extensive experiments on these algorithms under the different edge constraints to observe their applicability and the corresponding heterogeneity pattern.**

***Index Terms*—Federated Learning, Platform, Model Heterogeneity**

## I. Introduction

Federated learning (FL) is empowering multiple real-world applications such as Google's Keyboard [1] and medical image analysis [2], whose key idea is to employ a set of personal clients to collaboratively train a deep learning model with the orchestration of a central server. In recent years, conducting FL among edge devices has become a noteworthy trend [3], [4]. Under this condition, the typical FL assumption with respect to keeping an identical model architecture is hard to hold since edge devices are usually resource-constrained.

The research community has recognized this issue and introduced the concept of *Model-Heterogeneous Federated Learning (MHFL)* [5]. In this context, each client maintains a model that accounts for the device's resource constraints, and the federated learning process operates among these heterogeneous models. MHFL can offer the following benefits: (1) For resource-constrained devices, MHFL provides the flexibility to adaptively conduct FL under diverse resource constraints (e.g., computation, communication, and memory) by adjusting the model size. (2) For individual edge users with distinctive preferences, MHFL allows them to customize the model's structure and size. Overall, MHFL is well-suited for deployment on various edge devices.

In the past few years, a large number of papers have been published to improve the performance of MHFL [6], [7]. After exploration, we find that all of them are based on simple configurations (i.e., using the proportion of the original model to satisfy device constraints). However, they do not consider real-world device constraints. As shown in Table I, on two real edge devices (Jetson Nano, Jetson Orin NX), models generated by different heterogeneous methods with the same proportion exhibit significant differences in terms of various resources, particularly for the computational cost (training time) and the memory usage. Therefore, *we argue that the current comparisons of model-heterogeneous federated learning methods are not fair, and the community should rethink and re-evaluate model-heterogeneous federated learning under practical device constraints*.

To gain a fair and comprehensive performance comparison of diverse model-heterogeneous mechanisms for real-world constraints of edge devices, we construct the first system platform **PracMHBench**. PracMHBench defines three representative heterogeneity levels, each of which includes several corresponding MHFL algorithms. The platform also contains multiple model architectures and metrics, which are evaluated on various data tasks, spanning from computer vision (CV), natural language processing (NLP), and human activity recognition (HAR). Based on the benchmark, PracMHBench further build three typical edge-side constraint cases, which we call *computation-limited MHFL, communication-limited MHFL, and memory-limited MHFL*, and conducts experiments on them. For each limitation setting, we create it based on the collected statistics of practical edge devices, such that the results can benefit and guide the MHFL deployment in real-world resource-constrained scenarios.

Based on PracMHBench, we first perform extensive experiments on different model-heterogeneous algorithms under the specific device constraint to demystify the performance on different data tasks. Next, in terms of our evaluated metrics, we analyze the pattern of different heterogeneity algorithms and levels, aiming to explore the most suitable strategy for different resource-constrained scenarios. Finally, through the experiments, we summarize several insightful observations and conclusions (details in Section V).

This paper makes the following contributions:

- We construct a system platform PracMHBench to characterize and evaluate diverse model-heterogeneous algorithms under the practical resource-constrained edge devices. To the best of our knowledge, this is the first work that delves into the deployment of MHFL on real-world scenarios [1].
- We perform extensive experiments on PracMHBench with different heterogeneity levels and device constraints, and the results reveal a comprehensive landscape of current MHFL algorithms from a real system perspective.
- Based on the experiments, we summarize the insightful observations and practical implications that can benefit the practitioners and researchers targeting resource-constrained MHFL systems.

## II. Background and Related Work

**Model Heterogeneity in Federated Learning.** In recent years, Federated learning (FL) tailored for massive edge devices has garnered unprecedented attention [3], [4], [8]–[10]. Unlike traditional federated learning that only focuses on algorithm performance, federated learning in such scenarios pays more attention to the characteristics of heterogeneous devices. For example, training VGG16 requires over 8GB memory with a batch size of 8 [11], making it hard to

*corresponding author.

This work was partly supported by the National Natural Science Foundation of China (62302054) and CCF-Huawei Populus Euphratica Fund (System Software Track, CCF-HuaweiSY202410).

[1]https://github.com/9yc/PracMHBench

TABLE I

MODELS GENERATED BY DIFFERENT HETEROGENEOUS METHODS WITH THE SAME PROPORTION (I.E., X0.5 OF THE ORIGINAL MODEL) ON JETSON ORIN NX AND JESTON NANO. THE STATISTICS INCLUDE THE NUMBER OF PARAMETERS, TRAINING TIME (ONE ROUND) AND MEMORY USAGE. 'N' REPRESENTS THE JETSON NANO. 'O' REPRESENTS THE JETSON ORIN NX.

| Method | Model | Parameters(M) | Training Time(s) | Memory Usage(M) |
|---|---|---|---|---|
| SHeteroFL | ResNet101 (x0.5) | 10.66 | N:430.24 O:212.72 | 593 |
| DepthFL | ResNet101 (x0.5) | 10.29 | N:515.93 O:254.65 | 1220 |
| FedRolex | ResNet101 (*0.5) | 10.75 | N:465.17 O:233.56 | 780 |
| FeDepth | ResNet101 (*0.5) | 10.54 | N:450.64 O:222.35 | 631 |

deploy for many edge devices. The research community has proposed *model-heterogeneous federated learning (MHFL)* to address the issue. The primary aim is to enable each device to customize its model size while ensuring federation with other devices.

Achieving MHFL encompasses two categories of state-of-the-art techniques. The first category is sub-model partial aggregation [5], [6], [12]. Its fundamental idea involves extracting sub-models from a large model that adheres to the memory constraints of devices, followed by weight aggregation based on the positional information of neurons. The second category employs knowledge distillation to achieve federated aggregation [7], [13]–[15]. Its primary pipeline involves: (1) The server leverages a public dataset to compute the output logits of each model and performs logits aggregation; (2) The server sends the aggregated logits back to each client, thereby enabling them as the teacher model for the distillation process. While MHFL has been extensively studied, its level of heterogeneity and the performance of various methods under different real device constraints have not been thoroughly explored. Our work aims to fill this gap by constructing a comprehensive and practical platform for MHFL in resource-constrained edge systems.

**FL Platforms.** Benchmarking and evaluating the performance of FL is essential for practitioners to understand and employ associated algorithms and configurations effectively. At present, there are two main categories of platforms for FL. The first category comprises general FL platforms, which primarily aim to comprehensively assess the performance of various FL algorithms under different conditions [16]–[18]. The second category consists of specific FL platforms, tailored for in-depth exploration of certain characteristics within FL systems [19]–[21]. For example, to assess the data heterogeneity, Li *et al.* [20] proposed a comprehensive data partitioning strategy to cover typical non-IID (non-independently and identically distributed) data cases and evaluated existing FL algorithms under such settings. Furthermore, Yang *et al.* [21] developed an FL platform to explore the impact of heterogeneity on performance, particularly focusing on hardware and state heterogeneity. In our study, we pay more attention to the impact of *model heterogeneity* on FL performance and the practice of conducting diverse heterogeneous methods on resource-constrained edge devices, which is fundamentally different from previous research efforts.

## III. BENCHMARK CONSTRUCTION

**PracMHBench** is a platform aimed to characterize and evaluate diverse model heterogeneity algorithms on practical resource-constrained devices. In this section, we first describe its benchmark construction in detail. The concrete statistics are shown in Table II.

**Model Heterogeneity Levels.** Due to diverse resource constraint conditions, various clients in federated learning may equip models with different capacities or architectures. In our PracMHBench, we define and categorize the model heterogeneity mechanism in an FL system into three levels (as shown in Figure 2):

- *Width Heterogeneity*, where the models employed by clients share the same topology while exhibiting differences in the number of channels in each layer, resulting in heterogeneous models with various widths.
- *Depth Heterogeneity*, where the model topology employed by clients is also identical and the difference lies in the number of layers, which can be considered as the depth-level heterogeneity.
- *Topology Heterogeneity*, where clients employ models with entirely different topologies. For example, an FL system may consist of diverse model architectures, such as ResNet [22], EfficientNet [23], MobileNet [24], and GoogleLeNet [25], in order to satisfy various user preferences and requirements.

**Representative Heterogeneous Algorithms.** In terms of the aforementioned heterogeneity levels, we select several heterogeneous algorithms for each level to represent its performance. Specifically, for *width heterogeneity*, we adopt Fjord [12], SHeteroFL [5] and FedRolex [6], whose key settings are allocating sub-models with different widths to clients before conducting specifically designed federation schemes. In the case of *depth heterogeneity*, we select FeDepth [26], InclusiveFL [27] and DepthFL [28], which achieves depth scaling by pruning one or more of the top layers of the global model. Towards *topology heterogeneity*, FedProto [7] and Fed-ET [14] are used to conduct FL with the help of knowledge distillation among these heterogeneous architectures.

**Data Tasks.** In our platform, we target three areas: computer vision (CV), natural language processing (NLP), and human activity recognition (HAR), each of which is evaluated by two representative datasets. For CV, we utilize CIFAR-10 and CIFAR100 [29]. For NLP, we employ AG-News [30] and Stack-OverFlow [31], and for HAR, we choose HAR-BOX [32] and UCI-HAR [33].

**Model Architectures.** To evaluate the different levels of model heterogeneity, we select various model architectures for experiments. For width and depth heterogeneity, we follow the settings outlined in related works [5], [6], conducting experiments using four different widths/depths (100%, 75%, 50%, 25% ) of the ResNet-101 [22] on the CIFAR-100 dataset [29] and MobileNetV2 [24] on CIFAR-10 dataset [29]. Additionally, for NLP tasks, we employ a customized transformer model [34] on AG-news and ALBERT [35] on Stack Overflow. We partition them into various dimensions to meet different definitions of heterogeneity. For HAR tasks, we employ customized CNNs following the previous work [36] and partition it into various dimensions to meet different definitions of heterogeneity. Regarding topology heterogeneity, we conduct experiments using two model families with distinctive architectures on CV tasks, including the ResNet Family [22] (ResNet-18, ResNet-34, ResNet-50, ResNet-101) on CIFAR-100 [29], and the MobileNet Family [24] (MobileNetV2, MobileNetV3 small, MobileNetV3 large) on CIFAR-10 [29]. Considering NLP models, we choose the ALBERT Family [35] (ALBERT base, ALBERT large, ALBERT xxlarge) on Stack Overflow [31]. Additionally, we modified the structure of the customized CNNs for HAR tasks.

**Evaluated Metrics.** To provide a comprehensive evaluation, PracMHBench adopts the following four metrics: (i) *Global accuracy*, where we maintain a global dataset and the final federated model is evaluated on it. (ii) *Time-to-accuracy*, which is defined as the wall

TABLE II
STATISTICS OF THE PROPOSED PLATFORM. HERE 'HETERO' REFERS TO THE HETEROGENEITY LEVEL.

| Hetero | Algorithm | CV | | NLP | | HAR | |
|---|---|---|---|---|---|---|---|
| | | Model | Dataset | Model | Dataset | Model | Dataset |
| Width | Fjord | ResNet-101 MobileNetV2 Variants (100%,75%, 50%,25%) | CIFAR10, CIFAR100 | ALBERT, Customized Transformer Variants (100%,75%, 50%,25%) | Stack-Overflow, AG-news | Customized CNN | HAR-BOX, UCI-HAR |
| | SHeteroFL | | | | | | |
| | FedRolex | | | | | | |
| Depth | FeDepth | | | | | | |
| | InclusiveFL | | | | | | |
| | DepthFL | | | | | | |
| Topology | FedProto | MobileNet, Resnet Family | | Albert Family | | | |
| | Fed-ET | | | | | | |

TABLE III
EDGE DEVICES USED IN OUR PLATFORM CONSTRUCTION.

| Devices | Processors | GPU Memory |
|---|---|---|
| Jetson orin nx | 1024-core NVIDIA Ampere GPU | 16GB |
| Jetson tx2 nx | 256-core NVIDIA Pascal GPU | 4GB |
| Raspberry 4B | Broadcom BCM2711B0 quad-core A72 64-bit @ 1.5GHz CPU | no gpu |

clock time for training a deep learning model to reach a pre-set accuracy, aiming at measuring the training speed. (iii) *Stability*, where we record the accuracy of each device and compute the variance among them to show the stability for various heterogeneous models. (iv) *Effectiveness*, where we compare the final accuracy of MHFL algorithms with a simple resource-aware homogeneous baseline (i.e., training the smallest homogeneous model across all heterogeneous devices). The effectiveness is measured by the accuracy improvement. Our goal is to test whether heterogeneous models are actually needed (i.e., whether the improvement is greater than zero) and the benefit of conducting MHFL (i.e., the improvement value).

## IV. PRACTICAL DEVICE CONSTRAINTS

Given the above benchmark settings, we then manually build three typical resource-constrained cases based on real-world device properties (enumerated in Table III). As illustrated in Figure 3, we first build a model pool, where various deep learning models in the literature are measured and we pick out the suitable one in terms of the heterogeneity level and resource constraint. Our principle for selecting models from the model pool is to keep the constraint consistent for all methods to ensure fairness. In the following subsections, we describe each resource limitation on MHFL in detail. Note that for each specific limitation, other resources remain identical.

### A. Case I: Computation-Limited MHFL

**Description.** Limited computing power means that different devices may demonstrate varying levels of computational constraints. As shown in Table I, there exists a significant difference in computing power among real edge devices, leading to distinctive training speed. Under this condition, we define *Computation-Limited MHFL* as follows:

**Definition IV.1. (Computation-Limited MHFL)**. *Given the various computing capabilities of practical edge devices, the goal of computation-limited MHFL is to adapt the model size of its original version to ensure a same model training speed among devices for synchronous aggregation.*

**Setting.** To achieve such a scenario, we follow the IMA dataset settings [21], where they collected status from more than 1,000 devices (e.g., Samsung Note 10, Nexus 6, and Redmi Note 8) and recorded the real computational capabilities of these devices. We used the computing power of the IMA dataset to build the computation-limited scenarios. Specifically, we assign the diverse computational capabilities in the IMA dataset to the clients of the federated learning system as their training constraint, and select corresponding suitable models of different heterogeneity levels for each client from the model pool, with the help of the statistics of the training time for various models, which is provided by an AI Benchmark on various devices [37]. This helps ensure roughly equivalent training times for each device and achieve synchronous aggregation under a fixed submission deadline.

### B. Case II: Communication-Limited MHFL

**Description.** The limited communication bandwidth indicates that the speed of uploading/downloading model parameters is affected by the actual device hardware and network environment. In an FL system, a client with small communication bandwidth may spend a lot of time for uploading or downloading model parameters, resulting in its inability to upload model parameters in time for aggregation. The *Communication-Limited MHFL* can be defined as follows:

**Definition IV.2. (Communication-Limited MHFL)**. *Given the various communication bandwidth of different devices, the goal of communication-limited MHFL is to adjust the model size of its original version to ensure a same uploading/downloading speed among devices for synchronous aggregation.*

**Setting.** In the case of communication constraints, we control the communication time of each round in the federated learning system to a certain time (e.g., 200s) based on the real network bandwidth information provided by the IMA dataset. We then select appropriate models and corresponding quantities from the model pool to deploy different heterogeneity levels. Note that the parameter sizes corresponding to the various models under the different methods vary, and thus the model proportion of different sizes in the different methods also varies.

### C. Case III: Memory-Limited MHFL

**Description.** Unlike the above, limited memory capacity focuses more on the actual memory requirements of training each model on different devices. Limited memory may have a direct impact on whether a client can conduct training and changing the model size can directly satisfy a specific memory limit. In this case, we define *Memory-Limited MHFL* as follows:

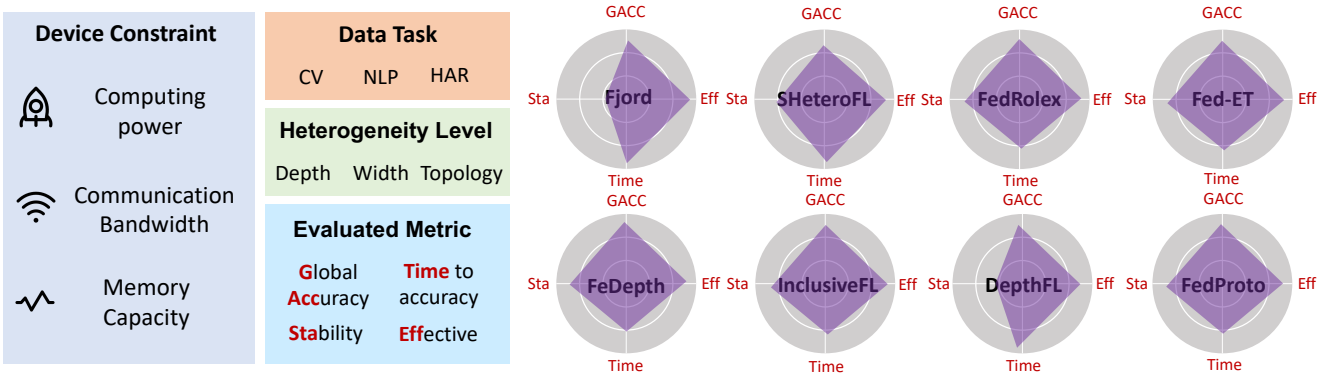



Fig. 1. Evaluation track of our platform PracMHBench. Here the information in radar charts is just for demonstration. The concrete performance will be compared in the subsequent experiments.

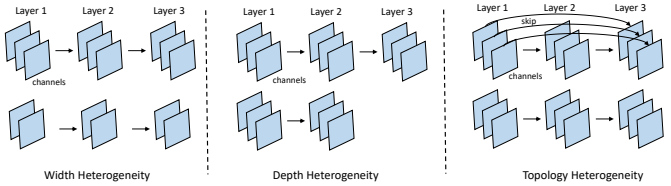



Fig. 2. Demonstration of different heterogeneity levels. Here 'skip' denotes the skip connection used in ResNets.

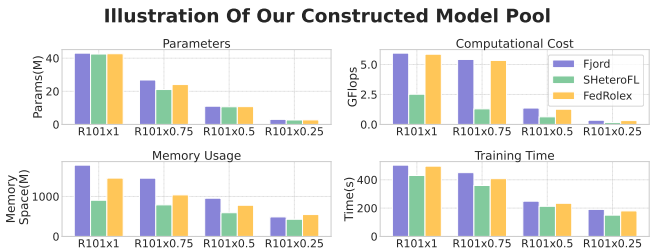



Fig. 3. Illustration of our constructed model pool. We use the tested system statistics (e.g., Parameters, GFlops) for three algorithms on Jetson Orin NX as an example. 'R' refers to ResNet.

**Definition IV.3. (Memory-Limited MHFL).** *Given the various memory capacity of different devices, the goal of memory-limited MHFL is to change the model size of its original version to ensure available local training for completing FL.*

**Setting.** We choose three typical GPU memory sizes of real devices: 16GB (e.g., Jetson Orin nx), 4GB (e.g., Jetson tx2 nx), and no GPU (e.g., Raspberry Pi). We then conduct experiments on real devices to observe whether the models in the model pool can be trained for federated learning. The largest trainable model is assigned to the client for meeting the corresponding memory size. In addition, the client proportion of different memory sizes is determined by the real-world proportion distribution depicted in [38].

## V. Evaluation and Analysis

Based on PracMHBench, we perform extensive experiments and analyze the results. The theme of this measurement is to quantitatively understand the performance discrepancy of different heterogeneity algorithms. As shown in Figure 1, the evaluation track includes: (1) selecting a targeted resource constraint; (2) conducting experiments on a variety of data tasks for different heterogeneity levels and algorithms; (3) recording and comparing the performance based on the evaluated metrics. In the following parts, we will describe the results of each resource constraint in detail. Note that some methods are not compatible with NLP tasks, and thus we omit the corresponding results. In addition, for the memory-limited scenario, we only focus on large CNN models (i.e., Resnet-101) and Transformer models (i.e., ALBERT) since most of the current edge devices can hold small models for HAR tasks.

Specifically, for CIFAR10, CIFAR100 and AG-News, we adopt IID (Independent and Identically Distributed) partition. For Stack Overflow, HAR-BOX and UCI-HAR, the dataset is partitioned over user IDs, making the dataset naturally non-IID distributed. The number of clients for CIFAR10, CIFAR100, AG-News, Stack-OverFlow, HAR-BOX, and UCI-HAR is 100, 100, 50, 500, 100, 30. The sampling ratio is 10%. All of the experiments are conducted for 1000 federated rounds to ensure convergence. Finally, we run each experiment 3 times and average them as the reported results.

### A. Results on Computation-Limited MHFL

**Global accuracy and time-to-accuracy.** The top row of Figure 4 shows the comprehensive results of global accuracy and time-to-accuracy. From the figure, we can observe that: (1) Overall, SHeteroFL and DepthFL achieve better performance on both global accuracy and time-to-accuracy. Regardless of the type of the data tasks, these two methods ensure relative superiority while Fjord and FedProto consistently exhibit the poorest performance across all data tasks. This suggests that under the computational constraint, the method's performance is unrelated to the type of data. (2) Although the two top-performing methods belong to different heterogeneity levels, we find that all depth-level methods (i.e., FeDepth, InclusiveFL, DepthFL) demonstrate favorable outcomes. This implies that adjusting the model size based on depth is a suitable choice in this scenario.

**Stability and effectiveness.** The bottom row of Figure 4 demonstrates the performance of stability and effectiveness. The following observations can be drawn: (1) Different from global accuracy and time-to-accuracy, which can identify the best algorithm for all data tasks, achieving a simultaneous satisfaction of stability and effectiveness is challenging on most datasets except for CIFAR-10, where FeDepth can be considered the best performer, which means that we can enhance the performance across nearly all devices. For other datasets, these two metrics tend to exhibit a trade-off, meaning that as stability increases, effectiveness tends to decrease. (2) For different heterogeneous levels, their performance on these two metrics does not follow any specific pattern and varies according to the differences in datasets.

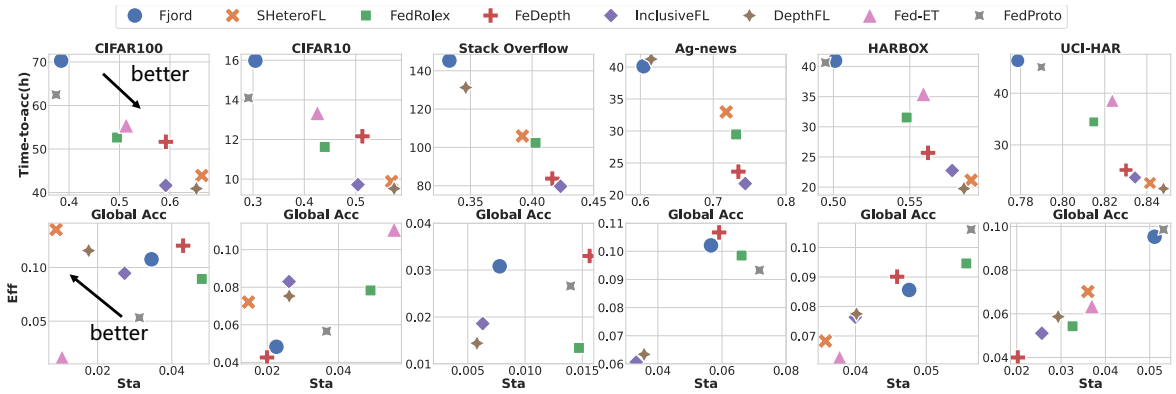



Fig. 4. Results on computation-limited MHFL. The top row refers to the comprehensive performance of global accuracy and time-to-accuracy. The bottom row represents the performance of stability and effectiveness.

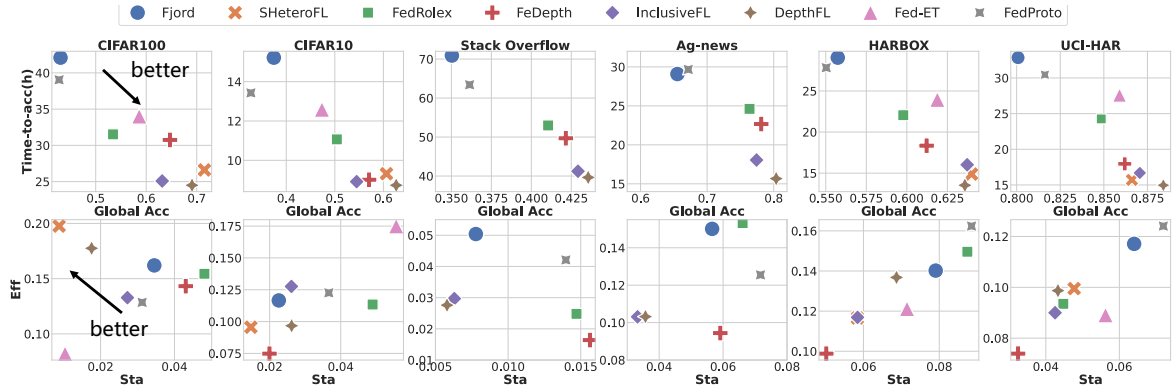



Fig. 5. Results on communication-limited MHFL.

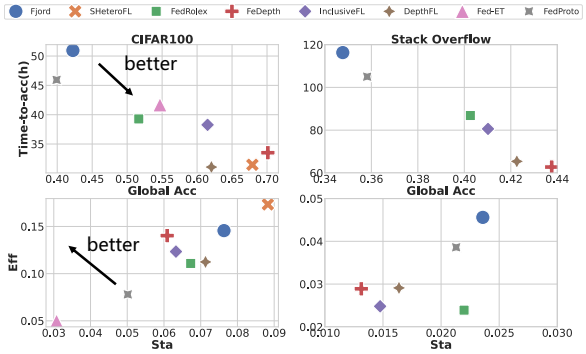



Fig. 6. Results on memory-limited MHFL.

> **Observation**: *In computation-limited MHFL, the superiority in terms of global accuracy and training speed remains a consistent pattern regardless of changes in data tasks. Among them, the methods belonging to the depth-level heterogeneity are relatively better than others. However, there is no one-size-fits-all algorithm that can achieve a desirable trade-off between the stability and effectiveness on diverse data tasks.*

### *B. Results on Communication-Limited MHFL*

**Global accuracy and time-to-accuracy.** The top row of Figure 5 shows the comprehensive comparison. We can draw the following conclusions: (1) Similar to the scenario of computational constraints, under communication bandwidth constraints, SHeteroFL and DepthFL still perform optimally, and this conclusion remains consistent in all data tasks. Besides, depth-level methods outperform other baselines. This indicates that the extent of changes in model size due to computational and communication constraints is not significant. Thus, the overall superiority of various model-heterogeneous algorithms remains unchanged. (2) Although the performance ranking has not changed, there is still a noticeable variation in the training speed gap among different methods. For example, on the Stack Overflow dataset, under the communication bandwidth constraint, the gap of time-to-accuracy (i.e., training speed) is significantly smaller than it is in the computational constraint (roughly 2X). On other datasets, the conclusion also holds. We believe this is because the communication constraint is more related to the size of the model parameters, which better aligns with the original design principle of the algorithms (i.e., controlling the proportion of model size). Therefore, each method can essentially follow the original settings, and the fluctuation in training speed will not be too significant compared to the computational constraint, which not only considers the model parameters but also takes into account the computational capabilities of different devices.

**Stability and effectiveness.** As depicted in the bottom row of Figure 5, we record the stability and effectiveness performance under the communication limitation. We can see that: (1) At a high level, it is hard to figure out which algorithm can simultaneously ensure stability and effectiveness across all data tasks. In other words, these two metrics do not follow any specific pattern. (2) We discovered that heterogeneous methods at the depth level perform poorly in terms of effectiveness. The reason may be twofold: on the one hand, architectures related to depth itself exhibit good performance (with high global accuracy as mentioned in the figure); on the other hand, as stated above, the communication constraint does not differ significantly from proportional splitting. Therefore, the overall state aligns well with the original design of the corresponding paper. Therefore, the improvement after heterogeneity is not as pronounced.

**Observation**: *Communication-Limited MHFL somewhat resembles the proportional splitting in existing literature, as it pays more attention to the number of model parameters. Therefore, the gaps among various algorithms, especially in terms of training speed, are not significant. Additionally, compared to other heterogeneous levels, although depth-level heterogeneity still holds an advantage in global accuracy and training speed, the performance improvement (i.e., effectiveness) brought by heterogeneity is relatively limited.*

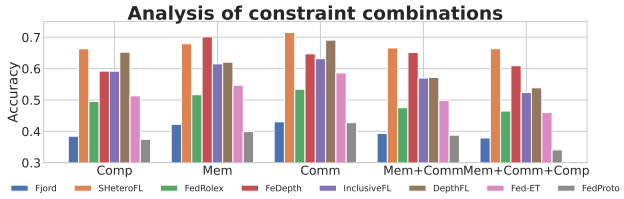



Fig. 7. Analysis of constraint combinations.

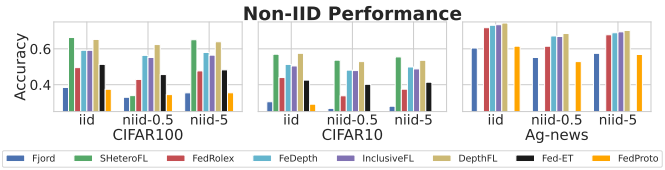



Fig. 8. Non-iid performance on the computation-limited scenario. Here '0.5' and '5' are the alpha of the non-iid dirichlet partition.

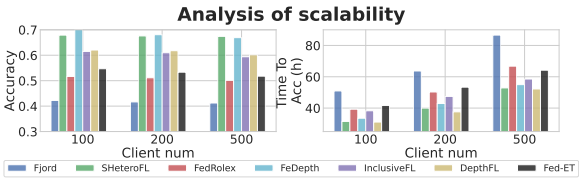



Fig. 9. Analysis of scalability.

### C. Results on Memory-Limited MHFL

**Global accuracy and time-to-accuracy.** Figure 6 shows the results. From the top row, the following phenomenon is noticeable: We *surprisingly* find that DepthFL, which performs well in the above two constraints, no longer exhibits superiority. Its global accuracy has significantly decreased, falling below the previously comparable SHeteroFL and inferior FeDepth. We attribute this to the fact that DepthFL itself requires a substantial amount of memory (shown in Table I). When memory is constrained, DepthFL needs to drastically reduce its size, leading to a decrease in performance. Instead, although FeDepth does not perform well under the computational and communication constraints, its small memory footprint (shown in Table I) allows it to accommodate larger models in this scenario, resulting in better performance.

**Stability and effectiveness.** We show the results at the bottom of Figure 6. From the figure, we find the following interesting observation: Compared to the two scenarios mentioned above, there is not much variation in the effectiveness trends of different algorithms. However, the conclusion regarding stability is almost reversed. For instance, in the case of CIFAR100, SHeteroFL exhibits the highest stability under the computation and communication constraints. However, for the memory constraint, its stability becomes the lowest among all methods. For Stack Overflow, Fjord demonstrates relatively high stability under the first two scenarios, but faced with memory limitations, its stability jumps to the lowest. We attribute this phenomenon to the significant difference between model parameters and actual device memory usage (as shown in Table I). Therefore, under the memory constraint, the model size of many methods may undergo substantial changes, leading to unstable performance.

**Observation**: *In practical memory-constrained scenarios, the performance of global accuracy and training speed, apart from the influence of heterogeneous levels and the methods themselves, is significantly affected by the actual memory footprint occupied by the models of different methods. Moreover, it does not exhibit a strong correlation with the types of data tasks. Additionally, concerning the stability, although it remains challenging to identify specific patterns, its trends are precisely opposite to the conclusions drawn under the computation and communication constraints.*

### D. Other Analysis

**Analysis of constraint combinations.** Here we test two types of combinations: communication+memory limited FL and computation+communication+memory limited FL on the CIFAR-100 dataset. Figure 7 demonstrates the accuracy results.We can observe that: Under the combination of communication+memory, the performance of all methods decreases compared with the single restriction and SHeteroFL performs best when various constraints are combined together due to its excellent partitioning method. SHeteroFL's slimmable partition method is well controlled in terms of calculation volume, memory consumption and communication volume, making it more adaptable to various restricted environments.

**Analysis of Non-IID Performance.** Figure 8 show the results of computational constraints under the non-iid scenario. From the figure, we can observe that although non-iid will degrade the performance, the conclusion is not changed, which means that existing algorithms is robust to the non-iid scenario.

**Analysis of scalability.** To demonstrate the scalability, here we conduct an experiment to evaluate the convergence-related metrics of various methods with different client numbers on the memory-limited MHFL setting and the CIFAR-100 dataset. As shown in Figure 9, scaling up leads to a decline in convergence performance, including both convergence effectiveness and speed. Among these methods, the width-level heterogeneity demonstrates relatively smaller performance losses, indicating better scalability.

## VI. Conclusion

In this work, we construct the first comprehensive platform for MHFL on practical resource constraints and conduct extensive measurements to quantitatively evaluate their performance. The results help reveal a complete landscape among diverse model heterogeneity levels and algorithms on real-edge devices. Based on the results, we summarize strong observations and implications that can be useful to FL developers and researchers of resource-constrained federated systems.

## References

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[2] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021.

[3] Bingyan Liu, Yifeng Cai, Ziqi Zhang, Yuanchun Li, Leye Wang, Ding Li, Yao Guo, and Xiangqun Chen. Distfl: Distribution-aware federated learning for mobile scenarios. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–26, 2021.

[4] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. Pyramidfl: A fine-grained client selection framework for efficient federated learning. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 158–171, 2022.

[5] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *ICLR*, 2021.

[6] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *NeurIPS*, 2022.

[7] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[8] Jaemin Shin, Yuanchun Li, Yunxin Liu, and Sung-Ju Lee. Fedbalancer: data and pace control for efficient federated learning on heterogeneous clients. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 436–449, 2022.

[9] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. Federated few-shot learning for mobile nlp. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–17, 2023.

[10] Bingyan Liu, Yifeng Cai, Hongzhe Bi, Ziqi Zhang, Ding Li, Yao Guo, and Xiangqun Chen. Beyond fine-tuning: Efficient and effective fed-tuning for mobile/web users. In *Proceedings of the ACM Web Conference 2023*, pages 2863–2873, 2023.

[11] Yongheng Deng, Weining Chen, Ju Ren, Feng Lyu, Yang Liu, Yunxin Liu, and Yaoxue Zhang. Tailorfl: Dual-personalized federated learning under system and data heterogeneity. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 592–606, 2022.

[12] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

[13] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.

[14] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *IJCAI*, 2022.

[15] Hong Zhang, Ji Liu, Juncheng Jia, Yang Zhou, Huaiyu Dai, and Dejing Dou. Fedduap: Federated learning with dynamic update and adaptive pruning using shared data on the server. *IJCAI*, 2022.

[16] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[17] Di Chai, Leye Wang, Liu Yang, Junxue Zhang, Kai Chen, and Qiang Yang. Fedeval: A holistic evaluation framework for federated learning. *arXiv preprint arXiv:2011.09655*, 2020.

[18] Yansong Gao, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit A Camtepe, Hyoungshick Kim, and Surya Nepal. End-to-end evaluation of federated learning and split learning for internet of things. *International Symposium on Reliable Distributed Systems (SRDS)*, 2020.

[19] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning*, pages 11814–11827. PMLR, 2022.

[20] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

[21] Chengxu Yang, Qipeng Wang, Mengwei Xu, Zhenpeng Chen, Kaigui Bian, Yunxin Liu, and Xuanzhe Liu. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *Proceedings of the Web Conference 2021*, pages 935–946, 2021.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[26] Kai Zhang, Yutong Dai, Hongyi Wang, Eric Xing, Xun Chen, and Lichao Sun. Memory-adaptive depth-wise heterogenous federated learning. *arXiv preprint arXiv:2303.04887*, 2023.

[27] Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Lingjuan Lyu, Hong Chen, and Xing Xie. No one left behind: Inclusive federated learning over heterogeneous devices. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3398–3406, 2022.

[28] Minjae Kim, Sangyoon Yu, Suhyun Kim, and Soo-Mook Moon. Depthfl: Depthwise federated learning for heterogeneous clients. In *The Eleventh International Conference on Learning Representations*, 2022.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[30] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[31] tff. Tensorflow federated stack overflow dataset, 2019.

[32] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. Clusterfl: a similarity-aware federated learning system for human activity recognition. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 54–66, 2021.

[33] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, page 3, 2013.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[36] Sannara Ek, François Portet, Philippe Lalanda, and German Eduardo Vega Baez. Evaluating federated learning for human activity recognition. In *Workshop AI for Internet of Things, in conjunction with IJCAI-PRICAI 2020*, 2021.

[37] Ai benchmark for different devices. https://ai-benchmark.com/ranking_deeplearning_detailed.html, 2021.

[38] ScientiaMobile. How much ram is in smartphones. https://www.scientiamobile.com/how-much-ram-is-in-smartphones/, 2022. Accessed: 2025-08-29.