FedDance: Efficient Participant Selection for Federated Learning in Highly Dynamic Environments

Abstract

Federated Learning (FL) is a rising distributed learning paradigm that facilitates multiple devices to jointly train a shared model. Given the presence of heterogeneous devices with distinct data distributions, it is critical to select an optimal subset of devices for engagement in the collaborative training process. However, the dynamic nature of FL, encompassing aspects like dynamic device availability and inherent training dynamics, significantly complicates participant selection, and current systems routinely fall short in adapting effectively to such dynamic environments.

This paper introduces FedDance, an efficient and intelligent participant selection framework tailored to tackle the dynamic nature in FL. FedDance employs a lightweight stochastic prediction method to anticipate the dynamic availability of each device. To address the diminishing marginal returns from frequently selected devices due to training dynamics, especially in later training stages, FedDance explicitly quantifies the marginal return of each local training process over time. Additionally, FedDance incorporates a lightweight model to characterize data heterogeneity, enhancing the advantages of managing device and training dynamics. Extensive experiments conducted on four A100 GPUs demonstrate that FedDance can substantially outperform the state-of-the-art systems in final model accuracy, while also providing a considerable advantage in computational overhead.

1 Introduction

FL has emerged as a key machine learning (ML) paradigm that harnesses the hardware capabilities and data from distributed devices to collaboratively train a shared model, offering significant scalability and robust data privacy protection [17, 25, 37, 68]. Modern FL systems often encounter two forms of heterogeneity. On the one hand, data is typically generated and stored locally under varying conditions, leading to substantial differences in the quantity and distribution of data across devices [50, 68]. On the other hand, devices exhibit diverse computing capacities and communication bandwidths [25, 39, 40]. Heterogeneity negatively impacts the final accuracy and the time required to achieve the target accuracy of the global model [7, 12]. Therefore, it is crucial to select an effective subset of devices as participants for joint training to address the heterogeneity, ultimately enhancing model accuracy and optimizing resource efficiency [9, 56].

Nevertheless, two significant challenges arise during the device selection process in FL. First, our examination of real-world device activity traces reveals that devices generally demonstrate highly dynamic availability during training due to fluctuations in battery levels and network connectivity [48, 63]. The accumulative online time of devices amounts to just 20.26% of the total time. This dynamic availability leads to a continually changing pool of online devices, each with notably diverse local data distributions [53], directly influencing the overall data distribution in each training round through global aggregation. Second, the local training process also exhibits a high degree of dynamism, with local models on different devices showing varying levels of incremental accuracy improvement over time. Without capturing the fine-grained dynamics of local model training during participant selection, the global model can easily suffer from diminishing performance gains or even accuracy degradation due to inappropriate aggregation. Our experimental findings indicate that selection strategies that prioritize high-availability devices alone perform even worse than a fully random scheme, showing a 12.5% decrease in global model accuracy.

Current participant selection methods for FL struggle to effectively tackle these challenges and are inadequate in dynamic environments characterized by data heterogeneity [34, 42, 51]. One such method, REFL [7], prioritizes devices with least availability to ensure wide participant coverage in collaborative training. Nevertheless, due to the oversight of data heterogeneity, it risks introducing significant bias to the global model. Our research shows that the final model accuracy of REFL lags behind that of a full random selection strategy by 2.0%. Additionally, REFL necessitates a tailored model trained on each local device to predict individual availability, leading to extra computational overhead beyond the regular shared model training. In contrast, Oort [29] meticulously selects participants that significantly contribute to improving global accuracy by favoring devices with highly diverse data distributions. However, a critical limitation of this approach is its inability to quantify the dynamic accuracy fluctuations in local models to assess the diminishing returns of participants. Participants with high utility, originating from hard-to-train samples [21, 38], may be overemphasized, leading to minimal contributions to the global model in later rounds. Consequently, Oort's final model accuracy falls behind by 3.1% compared to a fully random selection strategy.

To address these challenges, we introduce FedDance in this paper—a lightweight and effective participant selection framework specifically designed to address the dynamic nature of FL. FedDance incorporates three key components. First, it employs a statistical modeling technique based on a Poisson distribution to precisely estimate the dynamic availability of devices, without incurring additional consumption of the valuable computational resources of local devices. This is achieved by relying solely on the central server's ability to monitor binary time-series availability data during the check-in process. Second, FedDance characterizes data heterogeneity by analyzing the training loss produced during the training process, a method far more computationally efficient than existing approaches that rely on gradient estimation or evaluation on the complete local dataset. Third, FedDance explicitly tracks the accuracy enhancements on local devices over time to gauge the need for re-selecting a previous participant, with these improvements indicating a device's incremental value in collaborative training. Moreover, FedDance seamlessly integrates these three components into a unified metric to steer the participant selection process, while carefully balancing the exploration and exploitation tradeoff.

We have implemented FedDance on top of PySyft [2] and conducted comprehensive experiments to assess its performance on

four A100 GPUs across different training models and datasets. Our experimental findings demonstrate that FedDance accurately predicts the dynamic availability of all devices with over 90% precision, maintaining a computational overhead of each prediction below 10 microseconds. Furthermore, FedDance enhances model convergence speed by up to 1.39× and boosts global model accuracy by 18.23% compared to current state-of-the-art methods, while reducing device-side computation overhead by orders of magnitude, demonstrating exceptional efficiency. In summary, we have made the following contributions in this paper:

- We thoroughly examine the challenges posed by the inherent dynamic nature of FL in participant selection and illustrate that existing approaches fall short in effectively addressing these challenges.
- We design an intelligent participant selection scheme that systematically models a device's dynamic availability, data heterogeneity, and quantifies its diminishing returns in training accuracy throughout the dynamic training process, thereby sustaining the overall vitality.
- We develop an analytical framework to demonstrate the fast convergence of FedDance in the presence of dynamic device availability and non-IID data distributions.

2 Background and Motivation

In this section, we provide an overview of the core principles and current state of FL. We then delve into the challenges of deploying FL in dynamic real-world scenarios. Subsequently, we examine the shortcomings of current device selection strategies in addressing these challenges, which motivate us to design a new and efficient device selection strategy tailored for dynamic FL systems.

2.1 Basics of Federated Learning

FL is a distributed ML paradigm involving a powerful central server and dozens to even millions distributed devices [32]. These devices utilize their local data to cooperatively train a global model under the coordination of central server.

2.1.1 Characteristic of FL. In the typical FL training process, illustrated in Fig. 1, the workflow is divided into a series of rounds [36, 44]. At the beginning of each round, available online devices check-in with the central server to confirm their abilities of participation. Then, the central server samples a subset of these devices (① in Fig. 1) and broadcasts the global model to them (② in Fig. 1). Each participant performs local training to update its own model (③ in Fig. 1) and sends the model back to the central server (④ in Fig. 1). Thereafter, the central server aggregates these local updates to refine the global model (⑤ in Fig. 1).

Under FL, data on each candidate device is typically generated and stored locally under unique conditions, leading to variations in both quantity and distribution. For instance, in a healthcare application involving multiple wearable devices that collect patient data, the data from each device exhibits non-IID characteristics [24]. This disparity arises because health metrics like heart rate and step count vary significantly among individuals based on their lifestyles and fitness levels. Consequently, as each participant uses its local data to train a tailored model, considerable heterogeneity exists among the local models across participants.

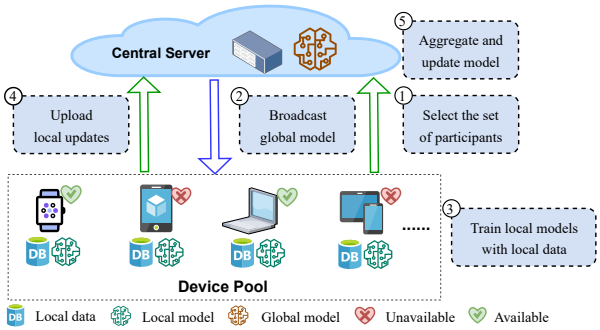


Figure 1: Training process in practical application of FL.

2.1.2 Problem formulation. To provide a clear interpretation of FedDance design, we first present the fundamental problem formulation of FL. Devices collaborate through a central server to solve the following distributed optimization problem:

$$\min_{\mathbf{w}} \left\{ F(\mathbf{w}) = \sum_{m=1}^{M} p_m F_m(\mathbf{w}) \right\}. \tag{1}$$

Here, M represents the total number of local devices, each with a local dataset $\mathcal{B}_m$ containing D_m samples. The proportion of data on the m-th device is expressed by $p_m = \frac{D_m}{\sum_{i=1}^{M} D_i}$. Additionally, $F_m(\mathbf{w}) = \frac{1}{D_m} \sum_{\xi \in \mathcal{B}_m} f(\mathbf{w}, \xi)$ denotes the local objective function of device m, where $f(\mathbf{w}, \xi)$ is the loss function involving sample ξ and model parameter vector $\mathbf{w}$.

Given the disparity in data distributions across heterogeneous devices, it's impractical to incorporate all these devices into joint training. Because the impact of some local updates on the global model performance is negligible or even detrimental. To conserve computing and communication resources, the central server selectively engages a subset of N devices from the pool of candidates during each round. Each participant performs local stochastic gradient descent (SGD) to update its tailored model. This paper designates the local SGD iterations with the index $t \geq 0$. Consequently, the participant set at iteration t is represented as $\mathcal{S}^{(t)}$. Since participant selection occurs only at the beginning of each training round, the set $\mathcal{S}^{(t)}$ remains unchanged for τ consecutive local update steps within the round. In other words, $\mathcal{S}^{(t+1)} = \mathcal{S}^{(t+2)} = \cdots = \mathcal{S}^{(t+\tau)}$ when $(t+1) \mod \tau = 0$.

2.2 Challenges in Participant Selection for FL

To motivate our system design, we first investigate the fundamental challenges that current FL system faces in terms of participant selection in dynamic environments.

C_1: Dynamic device availability. In FL, certain devices may be unable to engage in the entire training process due to fluctuations in battery status and communication network [46, 48]. To asses device availability in practical settings, we conducted an analysis using a public device activity trace [63], encompassing data from 136,000 mobile users over a week across various countries. This trace comprises millions of records detailing battery charging and WLAN connections. For analysis purposes, we segment each device's online duration into multiple availability periods, as depicted in Fig. 2. The results show that a device's active duration can be

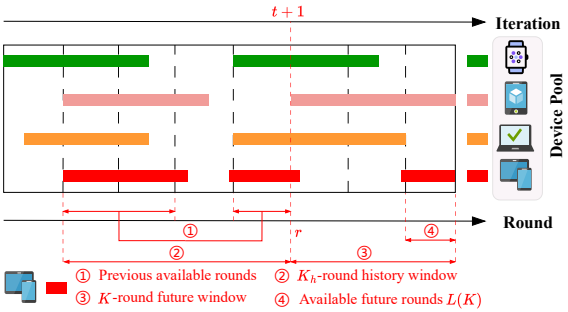
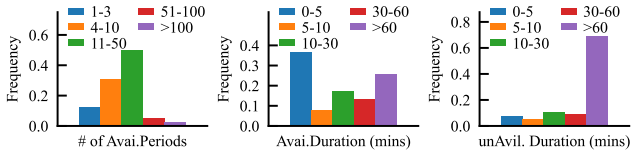


Figure 2: Illustration of devices' dynamic availability.



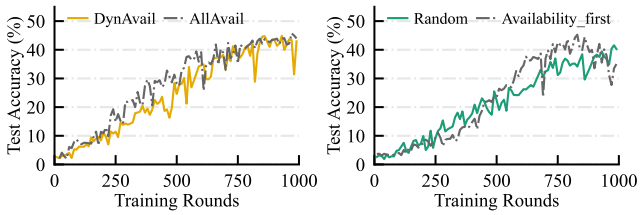
(a) Availability periods per device **(b) Duration of all availability periods** **(c) Gap of consecutive availability periods**

Figure 3: Availability characteristics of 136,000 users in a realistic device activity trace [63].

divided into dozens or even hundreds of these available periods, as illustrated in Fig. 3(a). However, most devices are accessible for only a few minutes within each available period, as shown in Fig. 3(b). Additionally, more than 65% of devices have time intervals exceeding one hour between consecutive availability periods, as demonstrated in Fig. 3(c). The collective online duration of all devices amounts to merely 20.26% of their total time.

This fluctuating availability during training results in a constantly changing pool of devices for selection, directly impacting the overall data distribution in each round [53]. Consequently, such dynamics can lead to the selection of inappropriate participants, negatively influencing both model accuracy and convergence rates [58, 66]. To validate this point, we conducted an experiment by randomly selecting 10 participants per round from a pool of 1,000 devices over 1,000 rounds in two conditions: 1) all devices are available (AllAvail); 2) devices' availability are determined by data extracted from the mentioned trace (DynAvail). Each participant trained a ResNet34 model [19] on the Google Speech dataset [60]. Figure 4(a) shows that dynamic device availability negatively impacts training efficiency, resulting in a 10.6% increase in the time required to achieve 40% test accuracy. This is because devices with dynamic availability may not be online for selection in every round, and even those that have been selected may not have the opportunity to participate in joint training again afterward. Consequently, their diverse data can introduce bias into the global model.

C_2: Diminishing marginal returns in model accuracy improvement during dynamic training. While certain devices consistently exhibit high availability, the performance enhancements of their customized local models gradually diminish if these devices are selected more frequently for participation. As a result, the weighted aggregation method in FL involving diverse devices [34]



(a) **(b)**

Figure 4: Impact of dynamic availability on test accuracy. (a) Dynamic availability degrades the global model accuracy compared to the scenario where all devices are available in each round. (b) The participant selection strategy that prioritizes high-availability devices leads to degradation in model accuracy in later rounds.

leads to diminishing marginal returns in accuracy improvement for the global model. Conversely, devices with low availability may hold significant improvement potential for their tailored models in later training rounds. This situation results in varying levels of contribution from these devices to the global model accuracy, especially during later stages of training.

To confirm this, we conducted an experiment using the same settings as described in C_1. In this study, devices with high availability were intentionally imbued with a greater degree of data heterogeneity in contrast to those with lower availability. As shown in Fig. 4(b), the strategy that exclusively prioritizes high-availability devices yields a final model accuracy even lower than randomly selecting devices during training, with a decrease of 12.5%. Therefore, simply favoring devices with high availability in participant selection proves inadequate. Notably, the global model accuracy from the availability-first approach exhibits a gradual decline after 800 rounds. Although high-availability devices can provide local training data in a relatively continuous manner over rounds, typically benefiting global model accuracy, as depicted in Fig. 4(a), over-reliance on the highly diverse data from these participants rapidly diminishes this advantage in the later stages of training. Consequently, given the varying availability of devices, this finding underscores the necessity of dynamically adjusting participant selection based on fluctuations in local model accuracy during the training process, a task that requires timely assessment. Without proper intervention, unnecessary local training and updates from less effective devices are included in joint training, wasting communication and computational resources [9].

2.3 Limitation of Existing Works

Substantial progress has been achieved in optimizing participant selection for FL under heterogeneous environments. Two standout works in this field include REFL [7], which can potentially address dynamic device availability uniquely, and Oort [29], an innovative study that assesses the impact of candidate devices on the global model. Despite their extensive efforts, these landmark works still face two fundamental limitations, as explained below.

Neglect of data heterogeneity evaluation in dynamic device availability scenarios. REFL utilizes a tailored prediction model for each device to predict its availability in a future timeframe, giving priority to the least available devices during participant selection in each round. However, REFL focuses exclusively on

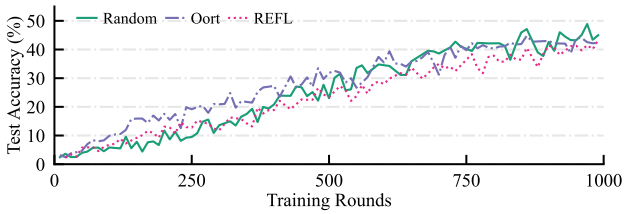


Figure 5: Test accuracy of various participant selection schemes in a scenario where the level of data heterogeneity for each device is inversely correlated with its availability.

availability, overlooking data heterogeneity [7]. As a result, REFL may consistently select devices with not only limited and fluctuating availability, but also subpar data. These intermittently usable devices may not engage in an sufficient number of training rounds, and the highly diverse local training data from these low-availability participants could easily introduce bias, potentially compromising the accuracy of the global model.

To emphasize this limitation, we conducted an experiment using the same setup described in § 2.2. Random selection from FedAvg [44] and REFL were employed to select devices for participating in the training process. The level of data heterogeneity for each device was set inversely proportional to its availability, highlighting the impact of diverse local data distributions on the global model performance. As shown in Fig. 5, the final accuracy of REFL notably lags behind that of Random selection by 2.0%. While REFL initially demonstrates advantages during the first 300 rounds by prioritizing less available devices to enhance resource diversity, the continuous integration of newly observed data from these low-availability participants—data that cannot be fully utilized— injects biases into the global model. Consequently, as training progresses, the global model accuracy under REFL gradually falls behind that of Random selection.

Inability to capture fine-grained training dynamics during participant selection. Oort employs a utility metric to select participants, which reflects the aggregate impact of a device's training data on the global model. Nevertheless, this metric fails to capture fine-grained variations in the accuracy of a device's customized model throughout dynamic training. As a result, Oort cannot effectively assess the diminishing returns from participants by quantifying the accuracy fluctuations of local models. Consequently, certain participants with high utility may be overutilized, offering limited contributions to the global model in later rounds.

To delve deeper into this limitation, we conducted an additional experiment under the same settings as previously outlined. In Fig. 5, it shows that Oort's final model accuracy falls 3.1% below that of Random selection. While Oort prioritizes devices that could potentially enhance the global model to sustain training effectiveness, its accuracy enhancement slows down quickly after 750 rounds. This deceleration arises from Oort's recurrent selection of certain high-utility devices, where the high utility values may stem from the persistent high loss of diverse data distributions within these devices, as per the utility function definition. And, their customized models experience diminishing marginal returns after an ample number of training rounds in the dynamic training process. From this perspective, Oort's inability to effectively link the local model dynamics of a device with its contribution to joint training results in devices with diminishing returns and excessive non-IID data characteristics continuing to be selected for training, leading to low efficiency.

3 FedDance Overview

To effectively address the challenges outlined in § 2.2, we proposes FedDance, a novel and lightweight participant selection scheme for FL. FedDance aims to achieve the following primary objectives: 1) enhancing final model accuracy and round-to-accuracy performance with devices that exhibit dynamic availability and high data heterogeneity; and 2) evaluating the effectiveness of each device during training without incurring unnecessary local computations.

3.1 Key Design Ideas

***I_1*: Accurate prediction of device availability.** To prevent the selection of devices with unstable availability that could impede model convergence, FedDance designs a model based on the Poisson distribution [22] to track device availability dynamics and predict future availability in a forward-looking manner using historical device registration data stored on the server. This approach runs entirely on the server side, enabling the central server to log the check-in process for each round and leverage its robust computational capabilities, thus requiring no additional computation from local devices and ensuring that no sensitive information is accessed.
***I_2*: Lightweight evaluation of data heterogeneity.** The adverse impact of data heterogeneity on model performance in dynamic environments, as depicted in Fig. 5, is clear. Thus, evaluating the data diversity of local devices in non-IID collaborative training is crucial. However, directly accessing the data distribution is infeasible due to privacy constraints [41]. To tackle this issue, FedDance employs the training loss produced during iterative local training for a lightweight assessment, using it to estimate a device's potential in upcoming training rounds without requiring extra computations. FedDance efficiently transmits the training loss at a minimal expense during the local model upload process.
***I_3*: Explicit quantification of participants' marginal returns.** The rate of accuracy improvement for the global model gradually diminishes over the training period, as depicted in Fig. 4(b). The quantified findings outlined in § 2.3 suggest that accuracy fluctuations serve as a more reliable metric for assessing each participant's influence on enhancing the global model accuracy. This is because a device with low accuracy but significant accuracy fluctuations may assume greater importance in the later stages of joint training. Hence, FedDance explicitly quantities a participant's training accuracy improvement across rounds to evaluate its marginal returns, sustaining training effectiveness amidst dynamic device availability and high data diversity. This approach capitalizes on training accuracy of the locally updated model without overburdening local computing resources.

3.2 System Architecture

Fig. 6 depicts the system architecture of FedDance system, comprising two essential components: **Local Controller** integrated into each accessible device and **Global Coordinator** situated in the central server. FedDance employs *utility* to signify the possible

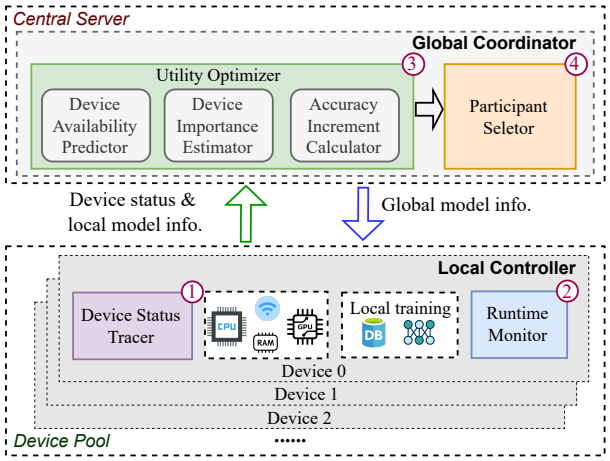


Figure 6: The system overview of FedDance.

impact of a candidate device on the overall performance of the global model.

Local Controller consists of two primary modules. *Device Status Tracer* (① in Fig. 6) sends device availability information to the server during the check-in process, while *Runtime Monitor* (② in Fig. 6) collects training accuracy for each round and training loss for each local iteration. After completing a round of local training, **Local Controller** transmits this data with the locally updated model to the central server.

Global Coordinator, functioning as a central component for handling feedback and conducting participant selection, consists of two modules: *Utility Optimizer* (③ in Fig. 6) and *Participant Selector* (④ in Fig. 6). *Utility Optimizer* includes three key parts: *Device Availability Predictor*, *Device Importance Estimator*, and *Accuracy Increment Calculator*. *Device Availability Predictor* assesses the future availability of each candidate device, on a scale from 0 to 1, by utilizing its individual check-in history with a device-specific Poisson distribution model. This availability value reflects the probability that a device will be active in at least one round within a K-round future window (as illustrated in Fig. 2). *Device Importance Estimator* evaluates data heterogeneity by considering the cumulative average training loss from a participant's local iterations in the preceding round. Meanwhile, *Accuracy Increment Calculator* computes a participant's accuracy improvement between its two consecutive engagement rounds based on training accuracy. It then averages these accuracy increments over a historical window to capture the participant's marginal return and assess the necessity of reselecting it. Once these factors for a participant are obtained, *Utility Optimizer* updates its utility accordingly. Subsequently, *Participant Selector* selects a subset of checked-in devices based on the utility metric provided by *Utility Optimizer*.

4 FedDance Design Details

In this section, we provide an in-depth explanation of the key components of FedDance. Specifically, we detail the design of three utility factors and describe the process of integrating them for online participant selection. Additionally, we present a convergence analysis of FedDance in scenarios characterized by dynamic device availability and heterogeneous data distributions across devices.

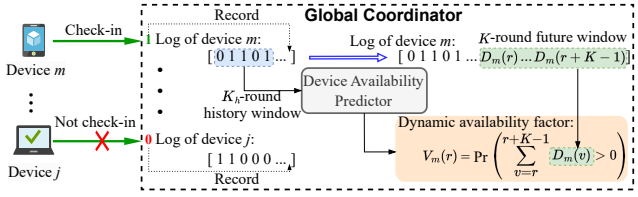


Figure 7: Workflow of Device Availability Predictor.

4.1 Device Availability Prediction

While REFL [7] also predicts devices' availability for participant selection, it requires each device to monitor a range of on-device events, such as charging status and screen locks. This data is then utilized to train a local Auto-Regressive Integrated Moving Average model to forecast future availability. However, due to variations in computing capabilities and storage capacities across devices, the practicality of REFL diminishes in real-world applications. Moreover, this approach compels the central server to send dedicated queries to retrieve prediction results from candidate devices, leading to increased communication costs and a more complex process.

FedDance adopts a distinctive strategy that capitalizes on the robust computational power and abundant storage of the central server. Specifically, within the FedDance protocol, the server is responsible for documenting the historical availability details of each local device during the check-in process. This historical log from the preceding round v, denoted as $D_m(v)$, consists of a single attribute: a binary value (0 or 1) indicating availability of the device during that specific round. Utilizing this information, *Device Availability Predictor* develops a customized statistical model for each local device to proactively predict its future availability. Notably, this method neither requires access to private data nor imposes any additional burden on the devices.

Given the significant fluctuations in device availability, as depicted in Fig. 3, estimating long-term stability offers a more precise approach than predicting immediate availability for the incoming round. Consequently, *Device Availability Predictor* utilizes a K-round future window (see Fig. 2), synchronized with the availability period's timescale, to assess the likelihood of the device being active in at least one round within that timeframe, as illustrated in Fig. 7. At the core of this estimation method is the concept that each K-round future window can serve as a good representation of the device's complete availability pattern, showcasing consistent statistical characteristics. In accordance with this principle, *Device Availability Predictor* employs a lightweight statistical modeling method based on the Poisson distribution to forecast the device's availability by leveraging historical statistics. The Poisson distribution is particularly well-suited for modeling time-series events [15]. Additionally, *Device Availability Predictor* integrates a history window of K_h rounds, adjusted based on round duration, to accurately depict the device's entire availability trace in the historical context while capturing its dynamic behavioral patterns.

Specifically, following the check-in process, the arrival rate of device m at current round r is updated as:

$$\lambda_m(r) = \frac{\sum_{v=r-K_h}^{r-1} D_m(v)}{K_h}, \tag{2}$$

where $D_m(v) \in \{0, 1\}$ represents the availability of device m at round v, with its check-in count serving as a proxy. Furthermore, the arrival rate can be readily updated as:

$$\lambda_m(r) = \frac{D_m(r-1) - D_m(r-1-K_h)}{K_h} + \lambda_m(r-1). \tag{3}$$

Under Poisson distribution, the probability that there will be k active rounds in total K future rounds, starting from the current round r, is expressed as:

$$\Pr\big(L_m(K) = k\big) = \frac{\big(\lambda_m(r) \cdot K\big)^k \exp\big(-\lambda_m(r) \cdot K\big)}{k!}, \tag{4}$$

where $k \in [K]$, and $L_m(K)$ is a random variable that denotes the count of active rounds within the K-round window. FedDance utilizes the *dynamic availability factor* to encapsulate the availability of each device, which is defined as the probability of having at least one active round, namely,

$$V_m(r) = \Pr\big(L_m(K) > 0\big) = 1 - \exp\big(-\lambda_m(r) \cdot K\big). \tag{5}$$

4.2 Device Importance Estimation

Explored in § 2.2 and § 2.3, it is crucial to assess the impact of a participant's data distribution in joint training [55]. Prior studies have indicated that larger gradient norms across participants' samples significantly influence the global model's performance [30]. Nevertheless, computing the gradient norm for each sample across all participants in every round is time-intensive, especially considering the diverse computing resources available on different devices. Research by [23] suggests a positive correlation between the loss value and the gradient norm to a certain extent. Oort [29] capitalizes on this by approximating the gradient norm through calculating the training loss for each sample in the local dataset. However, since local training typically involves only τ mini-batches, which are generally smaller than the complete local dataset, this method still demands additional computing resources due to the necessity of scanning the complete dataset.

For a mini-batch ξ_m randomly sampled from the local training dataset $\mathcal{B}_m$, the resulting average loss is unbiased [12]. This means that $\mathbb{E}\big[\mathbf{g}_m(\mathbf{w}_m, \xi_m)\big] = \frac{1}{|\mathcal{B}_m|} \sum_{\xi \in \mathcal{B}_m} f(\mathbf{w}, \xi)$, where $\mathbf{g}_m(\mathbf{w}_m, \xi_m) = \frac{1}{|\xi_m|} \sum_{\xi \in \xi_m} f(\mathbf{w}_m^{(t)}, \xi)$, and $\mathbf{w}_m^{(t)}$ represents the local model of device $\mathbf{w}_m^{(t)}$ at iteration t. Leveraging this property, at the current round r, *Device Importance Estimator* incorporates the training loss from mini-batches used during τ local iterations in the preceding round into the *importance factor* for device m, selected in the previous round. This factor quantifies the cumulative average training loss over local iterations as follows:

$$I_m(r) = \frac{1}{\tau |\xi_m^{(u)}|} \sum_{u=(r-1)\cdot\tau}^{r\cdot\tau-1} \sum_{\xi \in \xi_m^{(u)}} f(\mathbf{w}_m^{(t)}, \xi), \tag{6}$$

where $\xi_m^{(u)}$ represents a mini-batch comprising samples selected uniformly at random from participant's local dataset.

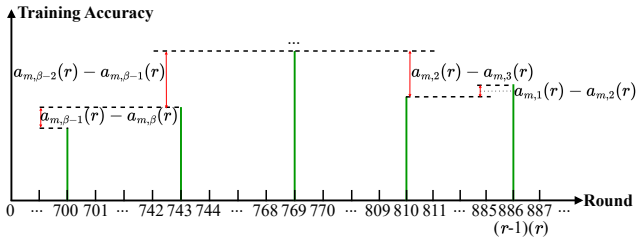


Figure 8: An example of accuracy improvement quantification.

An important advantage of this method is that *Device Importance Estimator* can obtain the loss as a byproduct of training, eliminating the need for a separate process to compute and record the gradient norm or calculate the loss for each sample in the local dataset.

4.3 Accuracy Improvement Quantification

Effectively quantifying each participant's marginal return is essential to enhance the global model accuracy in subsequent rounds. However, relying solely on loss may not provide an accurate measure of this return, as a participant with a high accumulated average training loss might contribute little to the global model if its data distribution is excessively skewed. In essence, loss alone fails to capture the nuanced training dynamics in collaborative learning. Conversely, changes in a participant's training accuracy reflect evolving performance more directly, providing insights into its contribution to joint training that the loss metric alone cannot reveal.

Building on this, FedDance calculates the increment in training accuracy between a participant's two consecutive rounds of involvement and averages these increments over a historical window during which the participant was engaged, to effectively quantify its past impact on joint training, as depicted in Fig. 8. This approach prioritizes devices with consistently high accuracy improvements rather than those with occasional accuracy peaks, ensuring a more reliable selection of valuable devices for collaborative training. To achieve this, at round r, FedDance uses *Accuracy Increment Calculator* within *Utility Optimizer* to compute the *accuracy improvement factor* for device m that is selected in the previous round, as defined by:

$$A_m(r) = \frac{a_{m,\beta}(r) - a_{m,1}(r)}{\beta - 1}, \tag{7}$$

where $\{a_{m,1}(r), ..., a_{m,\beta-1}(r), a_{m,\beta}(r)\}$ represents the local training accuracy of device m across the most recent β rounds in which it was chosen prior to the current round r. These metrics are recorded after each local training round. The parameter β is determined by the timescale of the availability period and round duration, guaranteeing a seamless reflection of the evolving dynamics within the training process.

4.4 Online Selection of High-utility Devices

FedDance selects participants based on their overall utility [29], with each device's utility dynamically reflecting its contribution to the global model throughout the training process. This utility

Algorithm 1 Participant Selection under FedDance

Input: $\mathcal{P}$-checked-in device set, $\mathcal{P}_{prev}$-checked-in device set of last round, $\mathcal{S}_{prev}$-participant set of last round, τ-steps of local update, N-participant set size
Output: The set of participants $\mathcal{S}$

/* Initialize global variables and variables of device m. */
1: $J_m \leftarrow 0$; $U_m \leftarrow 0$ ▷ Last involved round and utility
2: $a_m \leftarrow 0$; $R \leftarrow 0$ ▷ Training accuracy and round counter
3: $\mathcal{L}_m \leftarrow 0$ ▷ Accumulated average training loss
/* Participant selection for each training round. */
4: **Function** ParticipantSelector($\mathcal{P}$, $\mathcal{P}_{prev}$, $\mathcal{S}_{prev}$, τ, N)
5: $R \leftarrow R + 1$
6: J_m, $\mathcal{L}_m$, $a_m \leftarrow$ UpdateWithFeedback($\mathcal{P}_{prev}$, $\mathcal{S}_{prev}$)
7: **for** device $m \in \mathcal{S}_{prev}$ **do**
/* Update *importance factor*. */
8: $I_m \leftarrow$ UpdateImp(m, $\mathcal{L}_m$)
/* Update *accuracy improvement factor*. */
9: $A_m \leftarrow$ UpdateAcc(m, a_m)
10: **end for**
/* Calculate *average importance factor*. */
11: $\overline{I} \leftarrow \frac{1}{N}\sum_{m\in\mathcal{S}_{prev}} I_m$
/*Calculate *average accuracy improvement factor*. */
12: $\overline{A} \leftarrow \frac{1}{N}\sum_{m\in\mathcal{S}_{prev}} A_m$
13: **for** device $m \in \mathcal{P}_{prev} \backslash \mathcal{S}_{prev}$ **do**
14: **if** m was never selected in all previous rounds **then**
15: $I_m \leftarrow \overline{I}$, $A_m \leftarrow \overline{A}$
16: **end if**
17: **end for**
18: **for** device $m \in \mathcal{P}$ **do**
/* Update *dynamic availability factor*. */
19: $V_m \leftarrow$ UpdateAvi(m, R)
/* Calculate utility of online device. */
20: $U_m \leftarrow$ UpdateUtility(I_m, A_m, V_m)
/* Utility amplification. */
21: $U_m \leftarrow U_m\left(1 + \frac{\log_{10}(R+1)}{10(1+J_m)}\right)$
22: **end for**
23: Select devices with top-N utility from $\mathcal{P}$ to get the participant set $\mathcal{S}$
24: **return** $\mathcal{S}$

is calculated through a comprehensive integration of all the aforementioned factors, formulated as:

$$U_m(r) = V_m(r) \cdot I_m(r) \cdot A_m(r). \tag{8}$$

The detailed device selection procedure is described in Algorithm 1, where *Participant Selector* chooses devices with the top-N utility scores obtained from *Utility Optimizer* to form a set of N available devices within each round. In particular, for a local device m that was available in the prior round but consistently left unselected for joint training in all previous rounds, *Utility Optimizer* computes the average *importance factor* and *accuracy improvement factor* of participants chosen in the preceding round (Lines 11-15) to substitute for I_m and A_m. This ensures that, in the ongoing round, while leveraging high-utility devices, opportunities are also provided to potentially valuable devices that have yet to be explored.

Furthermore, this process integrates a mechanism to strike a balance between exploitation and exploration, which is crucial for improving the performance of the global model. Drawing inspiration from confidence bound in solving the multi-armed bandit problem [52], *Participant Selector* amplifies a device's utility if it has been neglected for a long time (Lines 21). This strategy gives a chance for devices that have been bypassed for some rounds since their last involvement, to be chosen anew, thereby maximizing the utilization of their training data.

Moreover, FedDance addresses a common issue encountered by existing device selection strategies that rely on utility-based metrics. These methods typically require an separate communication mechanism to collect data from local devices for utility estimation [7, 12]. For instance, REFL mandates a specific communication protocol to query each online device and retrieve device state statistics for participant selection before the common step of disseminating the global model in each round, thereby increasing communication costs. In contrast, FedDance calculates device utilities directly by leveraging the pertinent information embedded in local model updates from participants.

4.5 Convergence Analysis

Here, we theoretically demonstrate that FedDance converges to the global optimum at a liner rate.

4.5.1 Assumptions and definitions. First, we introduce the basic assumptions and definitions that underpin the convergence analysis in this paper.

Assumption 1. ***(Unbiased gradient)*** *For each uniformly sampled mini-batch ξ_m from $\mathcal{B}_m$, its stochastic gradient is unbiased, i.e., $\mathbb{E}[\mathbf{g}_m(\mathbf{w}_m, \xi_m)] = \nabla F_m(\mathbf{w}_m)$. Additionally, the variance of the stochastic gradient is bounded, meaning that $\mathbb{E}[\|(\nabla F_m(\mathbf{w}_m) - \mathbf{g}_m(\mathbf{w}_m, \xi_m))\|^2] \leq \sigma^2$ for $m = 1, 2, \cdots, M$.*

Assumption 2. ***(Bounded gradient)*** *The expected squared norm of the stochastic gradient is uniformly bounded, i.e., $\mathbb{E}[\|\mathbf{g}_m(\mathbf{w}_m, \xi_m)\|^2] \leq G^2$ for $m = 1, 2, \cdots, M$.*

Definition 1. ***(Degree of non-IID (data heterogeneity))*** *For the global optimum $\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w})$ and local optimum $\mathbf{w}_m^* = \arg\min_{\mathbf{w}} F_m(\mathbf{w})$, the degree of non-IID (heterogeneity) is defined as:*

$$\Gamma = F^* - \sum_{m=1}^{M} p_m F_m^* = \sum_{m=1}^{M} p_m (F_m(\mathbf{w}^*) - F_m(\mathbf{w}_m^*)) \geq 0, \tag{9}$$

where F^* and F_m^* denote the minimum values of $F(\mathbf{w})$ and $F_m(\mathbf{w})$, respectively. Γ is an intrinsic attribute of the distributed optimization problem in Eq. (1) and it is independent of the device selection strategy. A larger Γ indicates a higher degree of data heterogeneity. Conversely, if $\Gamma = 0$, it suggests that the global optimum aligns with the local optimum, indicating no bias in solutions due to the device selection strategy, as each device's data is IID distributed.

Definition 2. ***(Selection skew [12])*** *For any $m \in \mathcal{S}(\pi, \mathbf{w})$, the selection skew is defined as:*

$$\rho(\mathcal{S}(\pi, \mathbf{w}), \mathbf{w}') = \frac{\sum_{m\in\mathcal{S}(\pi,\mathbf{w})} q_m (F_m(\mathbf{w}') - F_m^*)}{F(\mathbf{w}') - \sum_{m=1}^{M} p_m F_m^*} \geq 0, \tag{10}$$

where the term $\rho(\mathcal{S}(\pi, \mathbf{w}), \mathbf{w}')$ reflects the skew between the selection strategy π and the scheme that selects all devices. $\mathcal{S}(\pi, \mathbf{w})$ represents the set of participants selected based on the global model $\mathbf{w}$ and the selection strategy π of FedDance. q_m is the weight of aggregation scheme $\mathbf{q}$ that satisfies $\sum_{m \in \mathcal{S}(\pi, \mathbf{w})} q_m = 1$. Due to the dependence of $\rho(\mathcal{S}(\pi, \mathbf{w}), \mathbf{w}')$ on the evolving global model parameters $\mathbf{w}$ and $\mathbf{w}'$, we introduce two associated metrics to establish a conservative error bound in the convergence analysis.

$$\hat{\rho} = \min_{\mathbf{w}, \mathbf{w}'} \rho(\mathcal{S}(\pi, \mathbf{w}), \mathbf{w}'), \ \widetilde{\rho} = \max_{\mathbf{w}} \rho(\mathcal{S}(\pi, \mathbf{w}), \mathbf{w}^*). \tag{11}$$

4.5.2 Convergence result. The key theoretical outcome is presented in the subsequent theorem, showcasing that FedDance achieves convergence to the global optimum at a linear rate, even amidst dynamic device availability and significant data heterogeneity—an aspect that has not been thoroughly examined in prior literature.

Theorem 1. *Under Assumptions 1 and 2, with a μ-convex and L-smooth function $F_m(\mathbf{w})$, a non-increasing learning rate η_t over t such that $\frac{\eta_{t_0}}{2} \leq \eta_t \leq \frac{1}{4L}$ for $0 \leq t - t_0 < \tau$, and $\gamma > 0$, $\beta = \frac{1}{\mu}$, the error after T iterations satisfies:*

$$\mathbb{E}\Big[F\Big(\sum_{m \in \mathcal{S}^{(T)}} q_m \mathbf{w}_m^{(T)}\Big)\Big] - F^* \leq \underbrace{\frac{8\Gamma(\widetilde{\rho} - \hat{\rho})}{3\hat{\rho}}}_{\text{Non-vanishing bias}} + \underbrace{\frac{L}{T + \gamma}\left(\gamma \|\overline{\mathbf{w}}^{(0)} - \mathbf{w}^*\|^2 + \frac{4(32\tau^2 N(N-1)G^2 + \sigma^2)}{3L\mu\hat{\rho}} + \frac{8\Gamma}{\mu}\right)}_{\text{Vanishing error}}. \tag{12}$$

Our theoretical analysis leverages online convex optimization techniques with constrained gradient descent. Taking into account the stochastic nature of device availability, we examine the expected outcomes over the dynamic candidate pool in each training round, aiming to robustly quantify the convergence of FedDance. Additionally, we evaluate the impact of data distribution diversity among local devices on overall model convergence using general averaging schemes, rather than simple uniform-weight averaging. For a detailed proof, please refer to [8].

We further explore the roles of the two terms in Eq. (12) to provide a comprehensive interpretation of Theorem 1.

Non-vanishing Bias. The first term, $\Psi(\widetilde{\rho}, \hat{\rho}) = \frac{8\Gamma(\widetilde{\rho} - \hat{\rho})}{3\hat{\rho}}$, represents the bias introduced by the selection strategy. Given the definitions of $\widetilde{\rho}$ and $\hat{\rho}$, we have $\widetilde{\rho} \geq \hat{\rho}$, which implies $\Psi(\widetilde{\rho}, \hat{\rho}) \geq 0$. This result also indicates that prioritizing devices whose data has not been sufficiently trained—leading to a smaller $\widetilde{\rho}$ and larger $\hat{\rho}$—reduces the value of $\Psi(\widetilde{\rho}, \hat{\rho})$, thereby decreasing the deviation between the global model parameters and the optimal parameters.

Vanishing Error. As training progresses, the vanishing error (the second term in Eq. (12)) approaches zero at a linear rate with respect to the number of completed training rounds. Given the substantial diversity in data distributions across local devices, it is evident that $\hat{\rho} > 0$. Consequently, the selection of devices with underutilized training data, which increases $\hat{\rho}$, further accelerates the rate at which the vanishing error diminishes.

5 System Implementation

We have implemented FedDance on top of PySyft [2]. Functioning as a distributed service, FedDance employs a standard communication library (such as XML-RPC [4]) to set up the communication link between the central server and clients.

Global Coordinator. Leveraging the PySyft framework, we implement two policies: *Utility Optimizer* and *Participant Selector* as core modules [1]. **Global Coordinator** listens for FL job submissions on the server-side. *Utility Optimizer* employs three long-running processes: *Device Availability Predictor*, *Device Importance Estimator*, and *Accuracy Increment Calculator*. These processes gather device status and local training feedback from participants via the PySyft API `model.get(participant_id)`. A queue is maintained to store the estimated utility for each device, which is then passed to *Participant Selector*. Subsequently, *Participant Selector* can invoke the clients with the top-N utility by using `model.send(participant_id)`.

Local Controller. We utilize the torch.multiprocessing library [6] to implement **Local Controller** on each client, which ensures that all clients remain isolated while sharing the same fundamental properties. Additionally, by employing the sys.monitoring library [3], each client's *Device Status Tracer* process collects device status information over time. Meanwhile, *Runtime Monitor* tracks each participant's training accuracy at the end of every round and monitors the training loss after each local iteration.

6 Evaluation

In this section, we provide a comprehensive evaluation of FedDance's participant selection strategy on global model performance and computing costs, highlighting its notable advantages over baseline methods, in scenarios with highly dynamic device availability and pronounced data heterogeneity.

6.1 Experimental Setup

Infrastructure. To evaluate the effectiveness of FedDance in real-world scenarios, our experiments were conducted on a cluster equipped with four NVIDIA A100 GPUs, each featuring 80GB of memory, and two Intel Xeon Gold 6342 CPUs, each equipped with 52 cores. The CUDA version is 12.4, and the NVIDIA driver version is 550.90.07. The training tasks are developed with PyTorch v1.13.1.

Training workloads. We selected four representative training workloads from the image and audio classification domains, with the corresponding models, datasets, and parameters outlined in Table 1.

Baselines. We compared FedDance with the following participant selection baselines:

- FedAvg [44]: In FedAvg, the central server randomly selects a subset of available devices to participate in joint training each round, without considering other factors.
- SAFA [61]: SAFA employs a compensatory selection strategy that prioritizes devices with less frequent participation to improve convergence rates under extreme conditions.
- REFL [7]: REFL focuses on the least available devices, where each device trains a local prediction model to estimate its probability of availability.

Table 1: Datasets and models used in the evaluation.

Dataset	Model	NS	#Labels	LS	BS	LR
EMNIST [13]	ResNet18 [19]	131,600	47	1	32	0.1
Tiny-ImageNet [14]	ShuffleNet [67]	110,000	200	4	20	0.2
Google Speech [60]	ReseNet34 [19]	104,667	35	1	32	0.075
CIFAR-10 [27]	ReseNet18	60,000	10	1	24	0.1

NS: Number of Samples. #Labels: Number of Labels. LS: Local Steps. BS: Batch Size. LR: Learning Rate.

- Oort [29]: Oort selects devices based on the utility metric that accounts for both the importance of training samples and local training time of each device, while completely ignoring dynamic device availability.

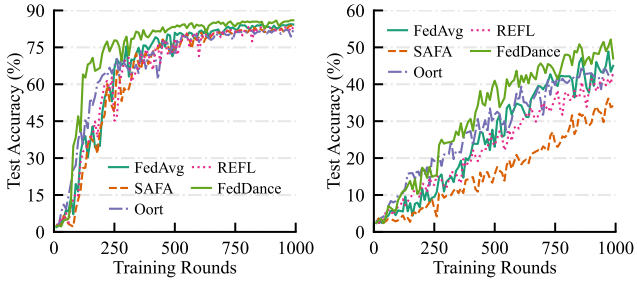
Parameters. The configuration parameters for FL were set to their default values as specified in the FedScale framework [28]. The parameter settings for the baselines followed those specified in the original literature [7, 29, 44, 61]. Additionally, common FL hyperparameters were kept consistent across all methods for a fair comparison. For FedDance, the future window K was set to 5, with the history window K_h and the accuracy improvement parameter β configured to 50 and 5, respectively.

Performance metrics. Two key metrics are round-to-accuracy performance [10, 35, 47, 61] and final model accuracy on the test set [29, 66]. Round-to-accuracy performance reflects data efficiency, measured by the number of rounds required to reach a target accuracy. Final model accuracy indicates the generalization ability of the trained model.

Data partitioning. To emulate heterogeneous data distributions in real-world scenarios, we sampled label ratios from a Dirichlet distribution, a prevalent data partitioning approach in FL [54, 62]. For each device m, using a symmetric parameter α, we sampled $p_{label}(m) \sim \mathtt{Dir}_M(\alpha)$ and distributed data to devices according to the corresponding label proportions. Notably, smaller α values indicate greater inter-device data heterogeneity [16]. Inspired by the works of [20, 33], we simulated realistic and diverse data distributions across devices using the following method: for half of the devices, α was uniformly sampled between 0.1 and 2, resulting in highly diverse data distributions; for the other half, α was uniformly sampled between 50 and 100, producing lower data diversity. This setup generated complex and varying data distributions across devices, enabling us to effectively assess the robustness of our approach. Moreover, for the Tiny-ImageNet dataset, which has a large number of labels, each device was assigned data containing 50 randomly selected labels from the total of 200, enhancing training efficiency.

Dynamic availability of devices. To emulate the dynamic availability of devices in real-world scenarios, we utilized a device activity trace that encompasses data from 136,000 mobile users over the course of one week across multiple countries [63]. This trace consists of millions of records, including information on battery charging and WLAN connections. In particular, a device is available when it has power and is connected to the network [33, 46].

Hardware performance of devices. Similar to the setup in REFL, clients' hardware performance was randomly assigned based on real device profiles from a popular AI Benchmark [5], which catalog training times for prevailing neural network models across a variety of Android devices (e.g., Samsung S23 Ultra and Xiaomi 14). These profiles include devices with at least 2GB of RAM and WiFi connectivity, reflecting typical configurations in FL scenarios.



(a) EMNIST (b) Google Speech

Figure 9: Round-to-accuracy performance.

Table 2: The required number of training rounds (speedup compared to FedAvg) to achieve the same target accuracy.

Dataset & Model	EMNIST —ResNet18		Tiny-ImageNet —ShuffleNet		Google Speech —ReseNet34		CIFAR-10 —ReseNet18	
TA	79%		14%		31%		45%	
	NR	Sp	NR	Sp	NR	Sp	NR	Sp
FedAvg	421	-	473	-	522	-	510	-
SAFA	552	0.76×	>1000	<0.47×	821	0.64×	811	0.63×
Oort	530	0.79×	629	0.75×	483	1.08×	562	0.91×
REFL	542	0.78×	538	0.88×	630	0.83×	>1000	<0.51×
FedDance	**303**	**1.39×**	**448**	**1.06×**	**420**	**1.24×**	**450**	**1.13×**

NR: Number of Rounds. Sp: Speedup. TA: Target Accuracy.

6.2 End-to-end Performance

We demonstrate the superior performance of FedDance through comparisons with baseline methods on various datasets and neural network models in scenarios characterized by high heterogeneity and dynamic device availability. In each round, 10 participants are selected from a pool of 1,000 intermittently available devices, with a round duration of 100 seconds. Our main findings are as follows:

- FedDance can improve the final model accuracy by up to 18.23%.
- FedDance can significantly accelerate model convergence by up to 1.39×.

In the following, we provide an in-depth analysis of their performance comparisons.

FedDance improves round-to-accuracy performance. We set the target accuracy for each task referring to related works [18, 29, 31] to investigate the convergence speed of all approaches. It is evident that FedDance achieves significant speedups in reaching the target accuracy, as illustrated in Fig. 9 and Table 2. Using FedAvg as a benchmark, FedDance demonstrates faster convergence: 1.39× on the Emnist dataset, 1.06× on the Tiny-ImageNet dataset, 1.24× on the Google Speech dataset, and 1.13× on the CIFAR-10 dataset. Fig. 9 highlights that FedDance maintains its advantages throughout the entire training process. The improvements in round-to-accuracy performance are attributed to the meticulously designed utility function. Specifically, the consideration of *importance factor* and *accuracy improvement factor*, with the latter seamlessly incorporating

Table 3: The test accuracy (%) after 1000 training rounds.

Dataset & Model	EMNIST —ResNet18	Tiny-ImageNet —ShuffleNet	Google Speech —ReseNet34	CIFAR-10 —ReseNet18
FedAvg	84.30	22.01	44.99	51.09
SAFA	83.49	7.35	34.14	45.71
Oort	82.18	18.52	42.35	44.28
REFL	83.26	20.80	43.33	49.07
FedDance	**85.99**	**25.58**	**46.97**	**52.67**

the dynamic characteristics of the local model during joint training, ensures that FedDance effectively selects devices that accelerate the convergence of the global model.

FedDance promotes final model accuracy. Upon convergence, as depicted in Table 3, FedDance achieves a 1.69%-3.81% higher final model accuracy on the Emnist dataset, a 3.57%-18.23% improvement on the Tiny-ImageNet dataset, a 1.98%-12.83% enhancement on the Google Speech dataset, and a 1.58%-8.39% increase on the CIFAR-10 dataset. These substantial improvements can be mainly credited to the thorough modeling of the impact of dynamic device availability. This prevents short-sighted selections that could introduce biases, lowering final model accuracy due to the inclusion of data from low-availability devices that cannot participate in enough training rounds. Furthermore, *importance factor* within the utility function encourages FedDance to select valuable devices that efficiently enhance model accuracy. In contrast, a careless and inattentive participant selection strategy may involve devices with low availability and subpar data, thereby undermining the performance of the global model.

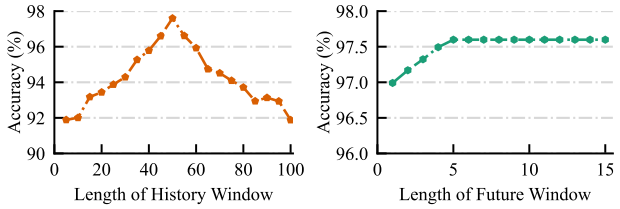
6.3 Microscopic Evaluation

In this part, In this section, we first test *Device Availability Predictor*, followed by an ablation study. Additionally, we analyze the sensitivity of the accuracy improvement parameter, β. The key points include:

- FedDance can predict device availability with consistently over 90% accuracy, while maintaining a computational overhead of less than 0.01 ms per prediction instance.
- FedDance demonstrates similar performance across a range of β values, consistently surpassing baseline methods in terms of final model accuracy.

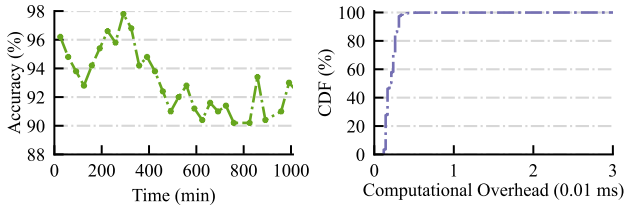
6.3.1 Performance of device availability predictor. To validate the effectiveness of *Device Availability Predictor* in complex scenarios with dynamic availability, we tested it using trace data from 500 devices. Each device's activity trace was leveraged to forecast its availability over a 1000-minute timeframe. Based on Eq. (5), we treated this as a binary classification problem, classifying a device as available if its predicted availability probability exceeded 0.5, and then compared this classification with the ground truth from its trace.

The 5-round future window, with each round lasting 100 seconds, aligns with the observation that most active periods of devices are under 10 minutes, as depicted in Fig. 3. Additionally, Fig. 10(a) indicates that a 50-round history window pairs best with *Device Availability Predictor* using this 5-round future window. Furthermore, Fig. 10(b) confirms the optimal pairing of the 5-round future



(a) Average prediction accuracy with a 5-round future window **(b) Average prediction accuracy with a 50-round history window**

Figure 10: Window sensitivity of *Device Availability Predictor*.



(a) Prediction accuracy of device availability over time **(b) CDF of computational overheads for all predictions.**

Figure 11: Performance of *Device Availability Predictor* with a 5-round future window and a 50-round history window.

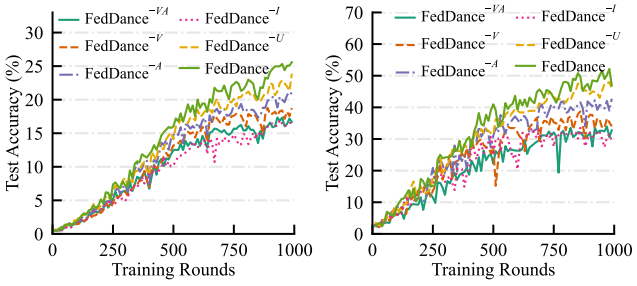
window with the 50-round history window. Consequently, using these windows, *Device Availability Predictor* achieve the highest average prediction accuracy over the 1000-minute test. Specifically, Fig. 11(a) demonstrates that the classification accuracy of availability for 500 devices consistently exceeds 90% throughout the test, despite dramatic fluctuations in the number of online devices.

Furthermore, Table 4 demonstrates that the average precision and recall over 1000 minutes are 97.60% and 97.09%, respectively. This reflects that only a few future-available devices are misclassified as future-unavailable, and vice versa. Fig. 11(b) highlights that the computational overhead for a single instance of device availability prediction is less than 0.01 ms. Therefore, given the high prediction accuracy and negligible computational overhead, the results of the device availability prediction are reliable for subsequent utility calculations and online device selection.

Table 4: Metrics of *Device Availability Predictor* with a 5-round future window and a 50-round history window.

Metric	Value
Average Accuracy	97.60%
Average Precision	97.09%
Average Recall	95.83%
Average F1 Score	96.22%

6.3.2 Ablation studies. We also quantified the impacts of different modules in *Utility Optimizer* and the confidence bound term in the utility function by comparing FedDance with several model variants: (1) FedDance^{-V}: *Device Availability Predictor* in *Utility Optimizer* is disabled, resulting in that certain devices with sharply varying availability enrolls in joint training; (2) FedDance^{-A}: *Accuracy Increment Calculator* is ablated, potentially leading to the selection of devices that offer negligible accuracy improvements, which do not benefit the convergence of the global model; (3) FedDance^{-I}:



(a) Tiny-ImageNet (b) Google Speech

Figure 12: Breakdown of round-to-accuracy performance.

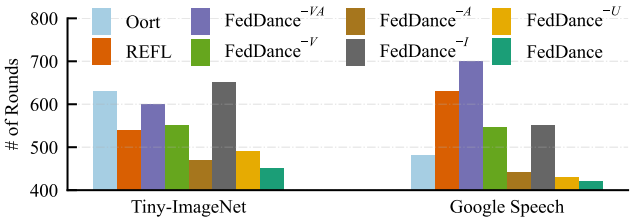
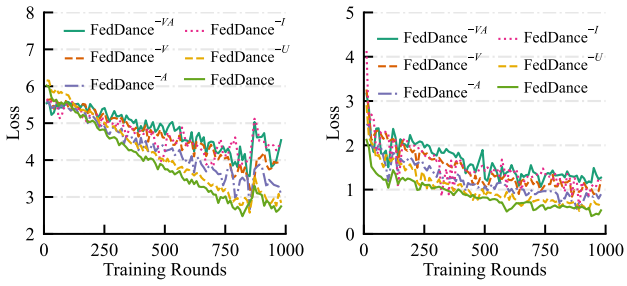


Figure 13: Number of rounds to reach the target accuracy.



(a) Tiny-ImageNet (b) Google Speech

Figure 14: Training loss analysis in the ablation study.

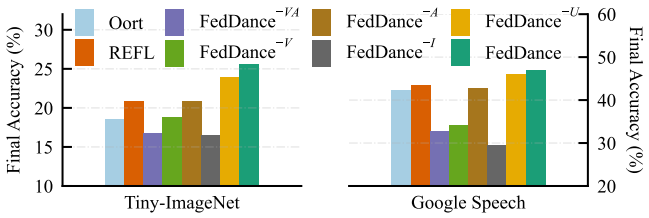


Figure 15: Breakdown of final model accuracy.

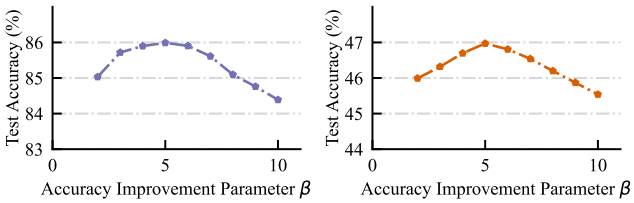
Device Importance Estimator is removed, increasing the likelihood of selecting devices with low-value data for training; (4) FedDance^{-VA}: Both *Device Availability Predictor* and *Accuracy Increment Calculator* are excluded; (5) FedDance^{-U}: The confidence bound term in the utility metric is removed, eliminating the extra privilege granted to devices that have been idle for a long time since their last participation.

Ablation study of round-to-accuracy performance. Fig. 12 presents the breakdown of round-to-accuracy performance, illustrating that FedDance benefits significantly from the carefully crafted modules in *Utility Optimizer*. The performance of FedDance^{-VA} is underwhelming. FedDance^{-V} improves model performance to some extent by integrating *Accuracy Increment Calculator*, which encourages the selection strategy to skip devices that contribute little to the joint training due to their minimal accuracy improvements. Meanwhile, FedDance^{-A} significantly improves training efficiency by fully accounting for device dynamics, preventing the introduction of negative bias from underutilized data contributed by devices with fluctuating availability. In contrast, the performance of FedDance^{-I} is inadequate, as it selects some devices with subpar data. Comparatively, FedDance^{-U} demonstrates superior performance, as it includes all the aforementioned modules but omits the confidence bound term in the utility function, which specifically prioritizes devices that have been idle after their last involvement. Overall, FedDance outperforms all these model variants by comprehensively considering all relevant elements, leading to a more well-reasoned participant selection.

We then adopted the target accuracy from Table 2 to examine the convergence speed of different methods. As shown in Fig. 13, FedDance exhibits a clear advantage over the model variants by incorporating all modules and the confidence bound term, achieving speedups of 1.31-1.44× over FedDance^{-VA}, 1.33-1.66× over FedDance^{-I}, 1.22-1.31× over FedDance^{-V}, 1.04-1.07× over FedDance^{-A}, and 1.02-1.09× over FedDance^{-U}. Furthermore, Fig. 14 reveals that FedDance converges to the minimum global training loss at a considerable rate. Therefore, FedDance can effectively accelerate the training process as analyzed in § 4.5.2.

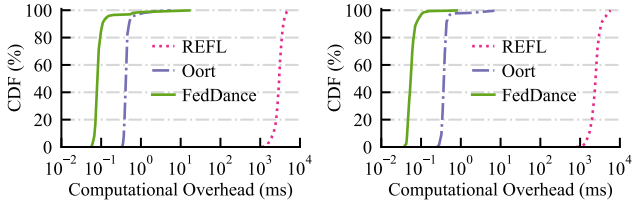
Ablation study of final model accuracy. Fig. 15 presents the breakdown of the final model accuracy. FedDance^{-I} exhibits subpar performance because it lacks evaluation of data value on local devices during joint training. In contrast to FedDance^{-I}, FedDance^{-VA} shows a modest improvement in accuracy, ranging from 0.16% to 3.25%, despite neglecting both the dynamic availability of devices and their past contributions in the absence of precise analysis. FedDance^{-V} achieves a more substantial accuracy gain of 2.24% to 4.52%, as it incorporates an *accuracy improvement factor* into the utility function, prioritizing devices that maintain training vitality, especially in the later stages of training. FedDance^{-A} demonstrates a stronger advantage, with accuracy improvements of 4.32% to 13.26%, as it models dynamic device availability to avoid short-sighted selections. Meanwhile, the performance of FedDance^{-U} is more prominent, delivering accuracy increases of 7.38% to 16.53%, since it just lacks the confidence bound term to preserve a balance between exploitation and exploration. Ultimately, FedDance outperforms all model variants, achieving an accuracy increase of 9.06% to 17.50%.

Impact of accuracy improvement parameter β. This parameter is to control the length of the history window used for quantifying accuracy improvements. As shown in Fig. 16, increasing β initially improves the final model accuracy, but beyond a certain threshold, it causes a decline. This is because, on the one hand, when β is small, the *accuracy improvement factor* only reflects the training accuracy increments from the most recent few rounds of participation, making it more susceptible to occasional accuracy peaks or short-term fluctuations between consecutive rounds. On the other hand, when β is large, the *accuracy improvement factor*



(a) EMNIST (b) Google Speech

Figure 16: The sensitivity of accuracy improvement factor.



(a) EMNIST (b) Google Speech

Figure 17: CDF of the computational overheads across all local devices during the participant selection stage over 1000 rounds.

incorporates increments from involved rounds that occurred much earlier, which may not accurately represent the current training dynamics of the local model. Nevertheless, a comparison of the final model accuracy in Table 3 and Fig. 16 reveals that FedDance, with β values varying within a certain range, consistently outperforms the baseline methods.

6.4 System Overhead

As shown in Fig. 17, FedDance markedly reduces the computational overhead on the local device during participant selection, achieving a nearly order-of-magnitude decrease compared to REFL and Oort. A closer examination reveals that the 90th percentile of computational overhead for all participants is 4 seconds for REFL, 0.5 ms for Oort, and 0.1 ms for FedDance. Similarly, the average computational overheads are 3 seconds, 0.47 ms, and 0.17 ms, respectively. Consequently, FedDance reduces the computational overhead by a factor of up to 3000 compared to REFL and by a factor of 3 compared to Oort, demonstrating its substantial efficiency gains.

The device-side computational overhead in FedDance arises from the device's *Runtime Monitor*, which only collects the training accuracy data within each round. In contrast, REFL requires each candidate device to train a local prediction model prior to the start of joint training, relying on various on-device events such as charging status and WLAN connections. Furthermore, during collaborative training, the parameters of each prediction model must be adjusted as the time window shifts. Similarly, Oort involves scanning each participant's dataset to handle training loss and update its utility through a relatively complex method, also demanding additional computational resources on the device-side.

7 Related Work

Participant selection is a critical problem in FL [11, 43, 49, 64, 65]. This section reviews three key aspects of participant selection, detailed below.

Prediction of dynamic device availability. A few studies have investigated the issue of dynamic device availability, which can adversely affect the performance of the global model. Power-of-choice [12] selects participants from available devices in each round based on a specific criterion, but it has only been tested in simple scenarios where each device comes online every other round. Moreover, it requires a separate process to gather state information from local devices before calculating the metric for participant selection. Similarly, REFL [7] predicts device availability by local prediction models. However, it introduces additional computational overhead for training these local models, and demands dedicated communication to retrieve these predictions.

Evaluation of data heterogeneity. Various research efforts have focused on addressing data heterogeneity [16, 29, 42, 45, 57]. Favor [59] estimates the data distribution of the differences in uploaded model weights and incorporates this information into the statistical utility to guide participant selection. TiFL [10] classifies devices into tiers based on their training performance and selects devices from the same tier in each round to address heterogeneity in resource and data quantity. AutoFL [26] designs a heterogeneity-aware participant selection strategy aimed at optimizing energy efficiency in FL. However, these approaches struggle in scenarios with dynamic device availability, where the pool of online devices is constantly evolving. This results in continuous shifts in the global data distribution across rounds due to the aggregation of diverse local data. Furthermore, they ignore accuracy fluctuations in local models, leading to short-sighted selections that degrade the long-term performance of global model.

Quantification of training dynamics. Several studies integrate device state information to maintain the vitality of the global model during the dynamic training process. FLOAT [24] leverages multi-objective reinforcement learning with human feedback. It incorporates device status into a device's reward function to automate the selection of optimization techniques, thereby improving participants' resource efficiency. FedRank [56] views participant selection as a ranking problem, and directly integrates the global model's test accuracy into the reward function. However, these approaches neglect local data distributions and fail to capture the fine-grained training dynamics. Additionally, the need for offline pre-training and human feedback renders these methods impractical and fragile in complex scenarios.

8 Conclusion

This paper thoroughly analyzes the challenges posed by the inherent dynamic nature of FL in participant selection, especially in the presence of significant data heterogeneity. To address these challenges, we propose FedDance, a robust participant selection framework for FL. To the best of our knowledge, FedDance is the first to simultaneously integrate three critical aspects of participant selection in FL: modeling of dynamic device availability, evaluation of data heterogeneity, and quantification of dynamic accuracy improvement. Moreover, our theoretical analysis and experimental results demonstrate that FedDance not only enhances model accuracy and accelerates model convergence, but also achieves this with minimal computational overhead on the device-side.

References

[1] 2019. Modules. https://docs.python.org/3/tutorial/modules.html.
[2] 2019. PySyft. https://www.measurementlab.net/tests/mobiperf/.
[3] 2019. sys.monitoring — Execution event monitoring. https://docs.python.org/3/library/sys.monitoring.html.
[4] 2020. xmlrpc — XMLRPC server and client modules. https://docs.python.org/3/library/xmlrpc.html/.
[5] 2021. AI Benchmark. https://ai-benchmark.com/ranking.html.
[6] 2023. Multiprocessing package - torch.multiprocessing. https://pytorch.org/docs/stable/multiprocessing.html.
[7] Ahmed M Abdelmoniem, Atal Narayan Sahu, Marco Canini, and Suhaib A Fahmy. 2023. Refl: Resource-efficient federated learning. In *Proceedings of EuroSys*.
[8] Anonymous. 2024. FedDance: Efficient Participant Selection for Federated Learning in Highly Dynamic Environments. https://www.dropbox.com/scl/fi/ae8t7qhw309yrumq9tx53/Technical-report.pdf?rlkey=mqz8m4sfd053nnql7bhtk2imf&st=xxag0179&dl=0
[9] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. 2022. Diverse client selection for federated learning via submodular maximization. In *Proceedings of ICLR*.
[10] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. 2020. Tifl: A tier-based federated learning system. In *Proceedings of HPDC*.
[11] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of AAAI*.
[12] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2020. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243* (2020).
[13] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *Proceedings of IJCNN*.
[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of CVPR*.
[15] Alex Dytso and H Vincent Poor. 2020. Estimation in Poisson noise: Properties of the conditional mean estimator. *IEEE Transactions on Information Theory* 66, 7 (2020), 4304–4323.
[16] Chun-Mei Feng, Kai Yu, Nian Liu, Xinxing Xu, Salman Khan, and Wangmeng Zuo. 2023. Towards instance-adaptive inference for federated learning. In *Proceedings of ICCV*.
[17] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. 2022. Blindfl: Vertical federated machine learning without peeking into your data. In *Proceedings of SIGMOD*.
[18] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of CVPR*.
[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR*.
[20] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).
[21] Justin M Johnson and Taghi M Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of big data* 6, 1 (2019), 1–54.
[22] Vladimir V Kalashnikov. 2013. *Mathematical methods in queuing theory*. Vol. 271. Springer Science & Business Media.
[23] Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*. PMLR, 2525–2534.
[24] Ahmad Faraz Khan, Azal Ahmad Khan, Ahmed M Abdelmoniem, Fountain, et al. 2024. FLOAT: Federated Learning Optimizations with Automated Tuning. In *Proceedings of EuroSys*.
[25] Md Saikat Islam Khan, Aparna Gupta, Oshani Seneviratne, and Stacy Patterson. 2024. Fed-RD: Privacy-Preserving Federated Learning for Financial Crime Detection. *arXiv preprint arXiv:2408.01609* (2024).
[26] Young Geun Kim and Carole-Jean Wu. 2021. Autofl: Enabling heterogeneity-aware energy efficient federated learning. In *Proceedings of MICRO*.
[27] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[28] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. 2022. Fedscale: Benchmarking model and system performance of federated learning at scale. In *Proceedings of ICML*.
[29] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient federated learning via guided participant selection. In *Proceedings of OSDI*.
[30] Bo Li, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. 2022. Partial variance reduction improves non-convex federated learning on heterogeneous data. *arXiv preprint arXiv:2212.02191* (2022).
[31] Bo Li, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. 2023. On the effectiveness of partial variance reduction in federated learning with heterogeneous data. In *Proceedings of CVPR*.
[32] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. 2022. PyramidFL: A fine-grained client selection framework for efficient federated learning. In *Proceedings of MobiCom*.
[33] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *Proceedings of ICDE*.
[34] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2021), 3347–3366.
[35] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of MLsys* (2020).
[36] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).
[37] Zengxiang Li, Zhaoxiang Hou, Hui Liu, Tongzhi Li, Chengyi Yang, Ying Wang, Chao Shi, Longfei Xie, Weishan Zhang, Liang Xu, et al. 2024. Federated Learning in Large Model Era: Vision-Language Model for Smart City Safety Operation Management. In *Companion Proceedings of the ACM on Web Conference 2024*. 1578–1585.
[38] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal Loss for Dense Object Detection. *arXiv preprint arXiv:1708.02002* (2018).
[39] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of CVPR*.
[40] Zelei Liu, Yuanyuan Chen, Yansong Zhao, Han Yu, Yang Liu, Renyi Bao, et al. 2022. Contribution-aware federated learning for smart healthcare. In *Proceedings of AAAI*.
[41] Zichang Liu, Zhaozhuo Xu, Benjamin Coleman, and Anshumali Shrivastava. 2024. One-pass distribution sketch for measuring data heterogeneity in federated learning. *Advances in Neural Information Processing Systems* 36 (2024).
[42] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems* 34 (2021), 5972–5984.
[43] Peihua Mai and Yan Pang. 2023. Privacy-preserving multiview matrix factorization for recommender systems. *IEEE Transactions on Artificial Intelligence* 5, 1 (2023), 267–277.
[44] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
[45] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of CVPR*.
[46] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. 2020. A first look at commercial 5G performance on smartphones. In *Proceedings of The Web Conference 2020*. 894–905.
[47] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. 2022. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3581–3607.
[48] Jianli Pan and James McElhannon. 2017. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal* 5, 1 (2017), 439–449.
[49] W Nicholson Price and I Glenn Cohen. 2019. Privacy in the age of medical big data. *Nature medicine* 25, 1 (2019), 37–43.
[50] Basheer Qolomany, Kashif Ahmad, Ala Al-Fuqaha, and Junaid Qadir. 2020. Particle swarm optimized federated learning for industrial IoT and smart city services. In *Proceedings of GLOBECOM*.
[51] Osama Shahid, Seyedamin Pouriyeh, Reza M Parizi, Quan Z Sheng, Gautam Srivastava, and Liang Zhao. 2021. Communication efficiency in federated learning: Achievements and challenges. *arXiv preprint arXiv:2107.10996* (2021).
[52] Yiwen Song and Haiming Jin. 2021. Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit. In *Proceedings of INFOCOM*.
[53] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2022), 9587–9603.
[54] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xinmei Tian, Tongliang Liu, Bo Han, and Xiaowen Chu. 2024. Fedimpro: Measuring and improving client update in federated learning. *arXiv preprint arXiv:2402.07011* (2024).
[55] Chunlin Tian, Li Li, Zhan Shi, Jun Wang, and ChengZhong Xu. 2022. Harmony: Heterogeneity-aware hierarchical management for federated learning system. In *Proceedings of MICRO*.
[56] Chunlin Tian, Zhan Shi, Xinpeng Qin, Li Li, and Chengzhong Xu. 2024. Ranking-based Client Selection with Imitation Learning for Efficient Federated Learning. *arXiv preprint arXiv:2405.04122* (2024).
[57] Saeed Vahidian, Mahdi Morafah, Chen Chen, Mubarak Shah, and Bill Lin. 2023. Rethinking data heterogeneity in federated learning: Introducing a new notion

and standard benchmarks. *IEEE Transactions on Artificial Intelligence* 5, 3 (2023), 1386–1397.

[58] Ewen Wang, Boyi Chen, Mosharaf Chowdhury, Ajay Kannan, and Franco Liang. 2023. Flint: A platform for federated learning integration. *Proceedings of MLsys* (2023).

[59] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing federated learning on non-iid data with reinforcement learning. In *Proceedings of INFOCOM*.

[60] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).

[61] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis. 2020. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans. Comput.* 70, 5 (2020), 655–668.

[62] Zikai Xiao, Zihan Chen, Liyinglan Liu, YANG FENG, Joey Tianyi Zhou, Jian Wu, Wanlu Liu, Howard Hao Yang, and Zuozhu Liu. 2024. FedLoGe: Joint Local and Generic Federated Learning under Long-tailed Data. In *Proceedings of ICLR*.

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

[64] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. 2023. PrivateFL: Accurate, differentially private federated learning via personalized data transformation. In *Proceedings of USENIX Security*.

[65] Jingwei Yi, Fangzhao Wu, Bin Zhu, Jing Yao, Zhulin Tao, Guangzhong Sun, and Xing Xie. 2023. UA-FedRec: untargeted attack on federated news recommendation. In *Proceedings of KDD*.

[66] Tuo Zhang, Lei Gao, Sunwoo Lee, Mi Zhang, and Salman Avestimehr. 2023. TimelyFL: Heterogeneity-aware Asynchronous Federated Learning with Adaptive Partial Training. In *Proceedings of CVPR*.

[67] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of CVPR*.

[68] Lei Zhao, Lin Cai, and Wu-Sheng Lu. 2023. Federated Learning for Data Trading Portfolio Allocation With Autonomous Economic Agents. *IEEE Transactions on Neural Networks and Learning Systems* (2023).