Coalitional FL: Coalition Formation and Selection in Federated Learning With Heterogeneous Data

Ning Zhang, Qian Ma, *Member, IEEE*, Wuxing Mao, and Xu Chen

***Abstract*—The model accuracy achieved by federated learning (FL) depends significantly on devices' data distributions. To improve the model accuracy of FL with heterogeneous data distributions on devices, existing works propose some device sampling methods for the central server, but face the problem that the selected devices may still have unbalanced data. In this article, we propose a novel coalitional FL framework for FL with heterogeneous data. Specifically, devices can cooperate and form device coalitions to reduce the data unbalancedness, and we formulate devices' interactions as a coalition formation game. Then the server selects an optimal subset of device coalitions to improve the model accuracy. Analyzing the coalition formation and selection framework is challenging since the relationship between model accuracy and data heterogeneity is not clear, and devices' coalition formation decisions and the server's coalition selection strategy are coupled in a highly non-trivial manner. We first derive a novel theoretical characterization of the relationship between model accuracy loss and data heterogeneity which follows an inverse function. With the novel theoretical relationship, we analyze devices' coalition formation game. We characterize the conditions under which the Nash stable partition exists, and propose an accelerated algorithm for devices to reach the Nash stable partition. For the server's device coalition selection problem, we show that the model accuracy loss depends on both data heterogeneity and the number of data samples of device coalitions in a non-monotonous way, and we propose a low-complexity algorithm for the server to select device coalitions efficiently. We conduct extensive simulations and show that our proposed coalition formation and selection framework reduces the data heterogeneity of selected device coalitions by up to 58.6% and increases the model accuracy by up to 6.8% compared with four existing benchmarks.**

***Index Terms*—Coalition formation game, data heterogeneity, device selection, federated learning.**

Manuscript received 26 June 2023; revised 2 February 2024; accepted 5 March 2024. Date of publication 14 March 2024; date of current version 3 October 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102700, in part by the National Natural Science Foundation of China under Grant 62372481 and Grant U20A20159, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011853 and Grant 2021B151520008, in part by Guangzhou Basic and Applied Basic Research Program under Grant 2024A04J6367, and in part by Shenzhen Science and Technology Program under Grant 202305063000008-2023A005, 20231121163344001. Recommended for acceptance by M. Zhang. *(Corresponding author: Qian Ma.)*

Ning Zhang is with the School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, Guangdong 518107, China, and also with the Research Institute of China Telecom Company Ltd., Guangzhou, Guangdong 510630, China (e-mail: zhangn87@mail2.sysu.edu.cn).

Qian Ma and Wuxing Mao are with the School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, Guangdong 518107, China (e-mail: maqian25@mail.sysu.edu.cn; maowx3@mail2.sysu.edu.cn).

Xu Chen is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong 510006, China (e-mail: chenxu35@mail.sysu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2024.3375325, provided by the authors.

Digital Object Identifier 10.1109/TMC.2024.3375325

I. Introduction

A. Background and Motivations

FEDERATED learning (FL) is a recently proposed distributed learning approach [1]. Compared with traditional machine learning frameworks which collect a large amount of data from devices and train AI models on a central server, FL can protect data privacy since it does not require devices to upload their raw data to the central server. The FL process involves many communication rounds between the central server and devices. In each round, as shown in Fig. 1(a), the central server selects a subset of devices and distributes the global model to selected devices. Each selected device then performs local training and uploads local model updates to the central server for model aggregation. The above interactions stop when reaching a predefined condition (e.g., the maximum number of communication rounds or a required model accuracy level).

The model accuracy achieved by FL depends significantly on devices' data distributions. Some works study FL with homogeneous data where devices have the same data distribution [2], and researchers mainly analyze the convergence performance of FL [3] and incentive mechanisms for FL to encourage more device participations [4]. In practice, however, devices usually have different data distributions. For example, when the central server trains a target recognition model in FL by recruiting mobile devices to perform local trainings on their photo albums, some devices have many photos of cars, some have many photos of pedestrians, and some have several photos of buildings. The data heterogeneity may compromise the model accuracy of FL.

For FL with heterogeneous data, characterizing the impact of data heterogeneity on model accuracy is of great importance. Some works analyze the model accuracy (or equivalently, the convergence performance) of FL with heterogeneous data, and introduce different metrics to quantify data heterogeneity, such as the weighted gradient diversities [5] and the local gradient variances [6]. However, the above metrics characterize the data heterogeneity by gradient, but cannot quantify the exact divergence of two data distributions. In this paper, we employ the earth mover's distance (EMD) metric originally proposed in [7] to quantify the data heterogeneity, which measures the minimum average distance required to move data from one distribution to the other distribution and evaluates the similarity between two probability distributions. Several works [7], [8] use EMD to

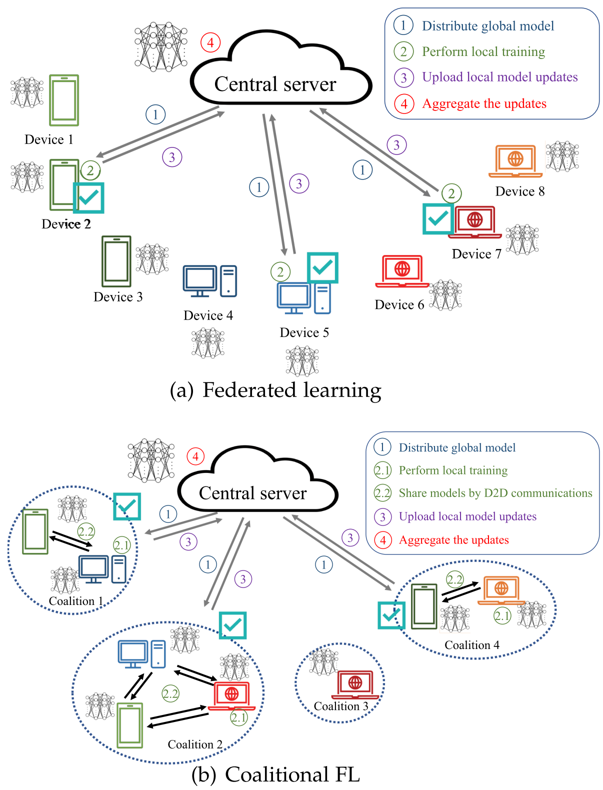


Fig. 1. Illustration of (a) FL and (b) coalitional FL.

characterize data heterogeneity in FL, but no work establishes the theoretical relationship between data heterogeneity (i.e., EMD) and model accuracy. Furthermore, most works characterize the model accuracy under the assumption that the central server selects devices uniformly at random or with probabilities being proportional to local data amounts [9]. In practice, however, the central server may select devices more intelligently to optimize its goal (e.g., maximizing the model accuracy), which brings more challenges to characterize the impact of data heterogeneity on model accuracy. This motivates us to address the first fundamental question in FL with heterogeneous data.

Key Question 1: How does data heterogeneity affect the model accuracy achieved by federated learning?

In FL with heterogeneous data, the interactions among devices also affect the performance of FL. Some works treat devices as competing entities, and the central server only selects a subset of devices for model trainings [10]. In this case, the selected devices usually still have unbalanced data, and unselected devices cannot contribute to FL model trainings [11]. In this paper, we propose a novel *coalitional FL* framework for FL with heterogeneous data, where individual devices with complementary data distributions can cooperate and form device coalitions to reduce the data unbalancedness, as shown in Fig. 1(b). In a device coalition, each device first performs local trainings, then shares its model updates with other devices in the coalition by D2D processes until reaching a consensus model [12]. When the central server selects a device coalition, a representative device in the device coalition uploads the consensus model to the central server for global aggregation. In coalitional FL, selected device coalitions usually have more balanced data, and more individual devices can contribute to FL model trainings by joining device coalitions, which can effectively improve the FL training performance.

When forming device coalitions, devices are rational and selfishly choose their coalition formation behaviors to maximize their own benefits. Specifically, to obtain a model with a high accuracy, the central server usually chooses and gives rewards to device coalitions with more balanced data. Therefore, a device tends to join a device coalition with many devices that can complement its unbalanced data to get rewards from the central server. This, however, will induce privacy loss to the device when exchanging model updates with other devices in the coalition [13]. Hence, each device needs to trade off between the reward and the privacy loss. In addition, devices' behaviors affect each other. For example, when a device joins a device coalition, it leads to a larger privacy loss to other devices in the coalition since the number of devices to exchange model updates increases. Furthermore, the reward received from the central server may also change, since the joining of a new device may lead to a coalition with more balanced data and hence a larger reward from the central server. Thus we model devices' selfish coalition formation behaviors as a coalition formation game. It is challenging to analyze the Nash stable partition of the game since the decisions of devices are coupled in a highly non-trivial manner. This motivates us to address the second fundamental question in our paper.

Key Question 2: How do devices selfishly choose their coalition formation behaviors and reach the Nash stable partition?

Considering devices' coalition formation behaviors, the server selects an optimal subset of device coalitions to maximize the model accuracy (or equivalently, to minimize the accuracy loss) in FL with heterogeneous data. Note that device coalitions are different in data heterogeneity and the number of local data samples, and the above two parameters of selected device coalitions affect the model accuracy in a non-monotonous way. For example, selecting a device coalition with a large number of local data samples and a small EMD can improve the model accuracy, while selecting a device coalition with a small number of data samples and a small EMD may reduce the model accuracy. When selecting device coalitions, the data heterogeneity and the number of data samples of device coalitions change simultaneously, which makes the coalition selection problem challenging to solve. In addition, the coalition selection problem is a large-scale integer programming problem, for which general solution methods are usually of high complexity. This motivates us to address the third fundamental question in our paper.

Key Question 3: How does the server select device coalitions to improve the model accuracy in FL with heterogeneous data?

B. Contributions

In this paper, we propose a novel *coalitional FL* framework for FL with heterogeneous data and study the coalition formation and selection problem. Specifically, devices with complementary data distributions form device coalitions to maximize their own payoffs selfishly, and the server selects an optimal subset of device coalitions to improve the model accuracy. We summarize our main contributions as follows.

- *Novel Performance Bound:* We derive a novel performance bound for FL with heterogeneous data which establishes the theoretical relationship between model accuracy loss and data heterogeneity. We are the first to show theoretically that the accuracy loss follows an inverse function regarding the EMD of selected devices.
- *Coalition Formation Game:* We model devices' selfish coalition formation behaviors as a coalition formation game. We show that when the server selects all device coalitions, the Nash stable partition always exists; and when the server selects a subset of device coalitions, the Nash stable partition exists if devices' privacy sensitivity is larger than a threshold. We further prove that the number of devices in a coalition has an upper bound, and we propose an accelerated algorithm for devices to reach the Nash stable partition.
- *Coalition Selection Algorithm:* It is challenging for the server to select the optimal subset of device coalitions since the model accuracy depends on selected device coalitions' data heterogeneity (i.e., EMD) and the number of local data samples in a non-monotonous way. We propose a low-complexity algorithm to guide the server to select device coalitions effectively by summarizing the two-dimensional parameter into a single metric.
- *Simulation Results:* We conduct extensive simulations on MNIST dataset. We show that our proposed coalition formation and selection framework can reduce the data heterogeneity of selected device coalitions by up to 58.6%. Compared with four existing benchmarks, our method can increase the model accuracy by up to 6.8%.

We organize our paper as follows. In Section II, we discuss related work. In Section III, we present the federated learning model and derive the performance bound. We propose the coalitional FL framework and model the interactions between the central server and devices in Section IV. We model and analyze devices' coalition formation game in Section V, and analyze the server's coalition selection problem in Section VI. We present simulation results in Section VII and conclude in Section VIII.

II. Related Work

Since McMahan et al. proposed FL in 2016 [1], FL has drawn many researchers' attention. In this section, we discuss related work regarding FL with heterogeneous data, clustering FL and coalitional FL, and device selection in FL.

A. FL With Heterogeneous Data

Data heterogeneity causes a slow convergence rate and a low model accuracy to FL, and many works aim to find promising solutions to tackle these problems. Li et al. propose FedProx which adds a proximal term to the objective function in FL to tackle heterogeneity, and prove the convergence of FedProx in both convex and non-convex cases [14]. Yang et al. propose a generalized FedAvg algorithm for FL with non-convex objectives on heterogeneous data, and the algorithm can achieve linear speedup for convergence under both full and partial participation [15]. Fallah et al. propose Per-FedAvg to train personalized models for clients with heterogeneous data which improves model accuracy [16]. Ding et al. show experimentally that a larger EMD causes a larger accuracy loss to FL and propose a contract based incentive mechanism to reduce the accuracy loss [8]. To tackle the data heterogeneity problem, some works generate fake data for classes with few data samples to achieve a more balanced data distribution and train a model with a higher model accuracy [7], [17], [18]. However, no existing work improves the model accuracy of FL with heterogeneous data by incentivizing devices with complementary data distributions to form coalitions, based on the theoretical relationship between data heterogeneity and model accuracy, which is the focus of our paper.

B. Clustering FL and Coalitional FL

Data heterogeneity has a detrimental impact on the performance of FL and many works apply the ideas of clustering and coalition formation to solve the problem. Some works group devices with the same or similar data into a coalition (or cluster) to reduce the FL training overhead or to train personalized FL models. Wang et al. in [19] and Lu et al. in [20] propose to cluster devices with similar data distributions and intelligently select devices in each cluster to participate in model trainings to improve the convergence performance of FL with heterogeneous devices. Long et al. propose to cluster clients with similar local models and learn multiple global models for better personalization to tackle the data heterogeneity challenge [21]. Arisdakessian et al. propose a trust-enabled coalitional game in FL where selfish clients form harmonious coalitions for local trainings and aggregations to mitigate the data heterogeneity problem [22]. Ng et al. propose a coalition formation framework where devices exchange intermediate parameters with neighbors to reduce communication latency in FL [23]. Wu et al. propose a personalized FL model where devices with similar data distributions form a coalition and perform personalized model aggregation to improve the personalized model accuracy [24]. However, none of the above papers studies the coalition formation of devices with complementary data to achieve more balanced training data, considering the server's intelligent coalition selection to improve the FL model accuracy.

C. Device Selection in FL

Device selection significantly affects the performance of FL and many researchers have studied this problem. Ma et al. propose a device scheduling scheme to select device groups based on group EMD to mitigate the FL performance degradation [25]. Guo et al. propose an adaptive reweighing scheme to select the optimal devices for FL according to the uplink transmit power values [26]. Pandey et al. propose a device selection approach for FL based on the contribution of devices to achieve faster convergence and better personalization [27]. However, most papers assume that the server selects devices based on a single metric (e.g., group EMD or processing capacity), which may not optimize the model accuracy of FL with heterogeneous data.

In this paper, we study the coalition formation and selection problem to improve the model accuracy of FL with heterogeneous data, considering the theoretical relationship between data heterogeneity and model accuracy.

III. Federated Learning Under Data Heterogeneity

In this section, we first introduce the framework of federated learning with heterogeneous data. Then we derive a novel theoretical characterization of the relationship between model accuracy loss and data heterogeneity for FL.

A. Federated Learning

In this subsection, we present the framework of federated learning with heterogeneous data. We first introduce the training goal of FL and the quantification of data heterogeneity. We then show the model update process.

We consider a central server who initiates a federated learning process and selects devices from the device set $\mathcal{K}$ (with $|\mathcal{K}| = K$) to train a global model represented by a parameter vector $\boldsymbol{\omega}$. Each device $k \in \mathcal{K}$ has a local dataset with n_k local data samples, and devices have different data distributions. The central server selects a subset $\mathcal{S} \subseteq \mathcal{K}$ of devices for model trainings. The training goal of federated learning is to obtain the optimal global model $\boldsymbol{\omega}^*$ which can minimize the global loss function [1]

$$F(\boldsymbol{\omega}) = \sum_{k \in \mathcal{S}} \frac{n_k}{\sum_{k' \in \mathcal{S}} n_{k'}} F_k(\boldsymbol{\omega}), \tag{1}$$

where $F_k(\boldsymbol{\omega})$ is the local loss function of device k. Specifically, we consider the multi-classification task [7],[1] and we denote a training data sample by $(\boldsymbol{v}, y)$, where $\boldsymbol{v}$ is the feature vector belonging to the compact space $\mathcal{V}$ and y is the label in the label set $\mathcal{Y} = \{1, 2, \ldots, C\}$. Devices have heterogeneous data distributions, and for device $k \in \mathcal{K}$, we denote the probability of one data sample having label i by $p_k(y = i)$. Under the parameter vector $\boldsymbol{\omega}$, the global model predicts the input sample $\boldsymbol{v}$ being class $i \in \mathcal{Y}$ with probability $f_i(\boldsymbol{v}, \boldsymbol{\omega})$ which is a mapping function depending on the learning model. In this case, we define the local loss function of device k by the widely used cross-entropy loss as follows [7]:

$$F_k(\boldsymbol{\omega}) = \sum_{i=1}^{C} p_k(y = i) \mathbb{E}_{\boldsymbol{v}|y=i}[\log f_i(\boldsymbol{v}, \boldsymbol{\omega})]. \tag{2}$$

Data heterogeneity significantly affects the training performance of FL, and we introduce earth mover's distance (EMD) to quantify each device's data heterogeneity. Specifically, the server has a population dataset [29] where the probability of one data sample having label i is $p(y = i)$. The EMD of a device quantifies the divergence between the data distribution of the device and that of the population dataset. We formally define the EMD [7], [8] of one device as follows.

Definition 1 (EMD): For each device $k \in \mathcal{K}$, the EMD measures the divergence between its data distribution and the data distribution of the population dataset, which is

$$d_k \triangleq \sum_{i=1}^{C} \|p_k(y = i) - p(y = i)\|. \tag{3}$$

Intuitively, when the device's data distribution is more biased from the population distribution, the EMD is larger.

We then define the weighted EMD of the set $\mathcal{S}$ of selected devices as follows.

Definition 2 (Weighted EMD): For the set $\mathcal{S}$ of selected devices, the weighted EMD is the weighted average EMD of all devices in set $\mathcal{S}$, which is

$$D_{\mathcal{S}} \triangleq \sum_{k \in \mathcal{S}} \frac{n_k}{\sum_{k' \in \mathcal{S}} n_{k'}} d_k. \tag{4}$$

We can see that the weighted EMD of selected devices depends on both the EMD and the number of local data samples. If a device has more data samples, its EMD has a larger impact on the weighted EMD.

We next describe the model training process of FL with heterogeneous data. Specifically, FL involves M communication rounds between the server and the selected devices. In each communication round $m \in \{1, 2, \ldots, M\}$, each selected device performs T steps of local model trainings $[(m-1)T + 1, mT]$, and we denote the m-th communication round by the shorthand notation $[m]$. At the beginning of round $[m]$, the server distributes the current global model $\boldsymbol{\omega}^{(fed)}_{[m-1]}((m-1)T)$ aggregated at the end of the previous round[2] (i.e., round $[m-1]$) to selected devices. Then each selected device $k \in \mathcal{S}$ performs T steps of local model trainings using the stochastic gradient descent (SGD) algorithm, where the local update at step $t \in [(m-1)T + 1, mT]$ is

$$\begin{aligned}\boldsymbol{\omega}_{[m],k}(t) &= \boldsymbol{\omega}_{[m],k}(t-1) \\ &\quad - \eta \sum_{i=1}^{C} p_k(y = i) \nabla_{\omega} \mathbb{E}_{\boldsymbol{v}|y=i}[\log f_i(\boldsymbol{v}, \boldsymbol{\omega}_{[m],k}(t-1))],\end{aligned} \tag{5}$$

and here η is the learning rate. At the end of round $[m]$, each selected device $k \in \mathcal{S}$ obtains the updated local model $\boldsymbol{\omega}_{[m],k}(mT)$ and uploads it to the server. Then the server aggregates selected devices' updated local models and updates the global model of round $[m]$ as

$$\boldsymbol{\omega}^{(fed)}_{[m]}(mT) = \sum_{k \in \mathcal{S}} \frac{n_k}{\sum_{k' \in \mathcal{S}} n_{k'}} \boldsymbol{\omega}_{[m],k}(mT). \tag{6}$$

The training iterations between the server and the selected devices repeat for M communication rounds, and at the end of the FL training process, the server obtains the global model $\boldsymbol{\omega}^{(fed)}_{[M]}(MT)$. Therefore, we can calculate the model accuracy loss of FL with heterogeneous data as $F(\boldsymbol{\omega}^{(fed)}_{[M]}(MT)) -$

[1]Multi-classification tasks have a wide range of practical applications, such as fingerprint recognization, face recognization, and disease diagnosis. Many works focus on multi-classification problem where data heterogeneity is a key concern [28].

[2]At the beginning of round [1], the server distributes the initial global model $\boldsymbol{\omega}^{(fed)}_{[0]}(0)$ to selected devices.

$F(\boldsymbol{\omega}^*)$, where $F(\boldsymbol{\omega}_{[M]}^{(fed)}(MT))$ and $F(\boldsymbol{\omega}^*)$ are the global losses under $\boldsymbol{\omega}_{[M]}^{(fed)}(MT)$ and the optimal global model $\boldsymbol{\omega}^*$ respectively [30]. In the next subsection, we analyze the impact of data heterogeneity on model accuracy loss.

B. Performance Bound

In this subsection, we analyze the model accuracy loss of FL with heterogeneous data. We first introduce three widely adopted assumptions on the loss function, and then derive a novel performance bound which establishes the theoretical relationship between model accuracy loss and data heterogeneity.

We first introduce the three typical assumptions of the loss function [31].

Assumption 1. ([31]): The local loss function $F_k(\cdot)$ of each device $k \in \mathcal{K}$ satisfies that:

- $F_k(\cdot)$ is β-smooth, i.e., $\|\nabla F_k(\boldsymbol{\omega}) - \nabla F_k(\boldsymbol{\omega'})\| \leq \beta\|\boldsymbol{\omega} - \boldsymbol{\omega'}\|$ for any $\boldsymbol{\omega}, \boldsymbol{\omega'}$.
- $F_k(\cdot)$ is L-Lipschitz, i.e., $\|F_k(\boldsymbol{\omega}) - F_k(\boldsymbol{\omega'})\| \leq L\|\boldsymbol{\omega} - \boldsymbol{\omega'}\|$ for any $\boldsymbol{\omega}, \boldsymbol{\omega'}$.
- $\nabla_\omega \mathbb{E}_{\boldsymbol{v}|y=i}[\log f_i(\boldsymbol{v}, \boldsymbol{\omega})]$ is $\lambda_{\boldsymbol{v}|y=i}$-Lipschitz for each label $i \in \mathcal{Y}$, i.e., for any $\boldsymbol{\omega}, \boldsymbol{\omega'}$, $\|\nabla_\omega \mathbb{E}_{\boldsymbol{v}|y=i}[\log f_i(\boldsymbol{v}, \boldsymbol{\omega})] - \nabla_\omega \mathbb{E}_{\boldsymbol{v}|y=i}[\log f_i(\boldsymbol{v}, \boldsymbol{\omega'})]\| \leq \lambda_{\boldsymbol{v}|y=i}\|\boldsymbol{\omega} - \boldsymbol{\omega'}\|$.

Then we derive a novel performance bound for FL which characterizes the theoretical relationship between model accuracy loss and data heterogeneity.

Theorem 1: Under Assumption 1, when $\eta < \frac{2}{\beta}(1 - \frac{LGD_{\mathcal{S}}}{(T-1)\phi\epsilon^2})$, the model accuracy loss of FL with heterogeneous data satisfies

$$F(\boldsymbol{\omega}_{[M]}^{(fed)}(MT)) - F(\boldsymbol{\omega}^*) \leq \frac{1}{M\eta(\phi(T-1)(1-\frac{\beta\eta}{2}) - \frac{LG}{\epsilon^2}D_{\mathcal{S}})}. \quad (7)$$

Proof: See Appendix A, available online in the supplementary material. □

Here ϕ, ϵ, and G are constants depending on the centralized machine learning process, which is independent of the FL process, and we relegate detailed discussions to Appendix A, available online due to page limit. Theorem 1 shows that the model accuracy loss $F(\boldsymbol{\omega}_{[M]}^{(fed)}(MT)) - F(\boldsymbol{\omega}^*)$ has an upper bound which is an inverse function of the weighted EMD $D_{\mathcal{S}}$ of selected devices.[3] We can see that when selected devices have a larger weighted EMD, the performance bound of FL with heterogeneous data is larger. Our analysis is consistent with the experimental results in [8], and we are the first to establish the theoretical relationship between model accuracy loss and data heterogeneity (i.e., the weighted EMD) for FL. Based on the performance analysis, the server needs to carefully choose an optimal subset of devices to minimize the model accuracy loss. In the next section, we propose a coalitional FL framework and analyze the interactions between the server and devices.

IV. Coalition Formation and Selection in FL

In this section, we propose the coalitional FL framework for FL with heterogeneous data. We first present the interactions between the server and devices. Then we model devices' coalition formation behaviors and the server's coalition selection problem, respectively.

A. Interactions Between the Server and Devices

Given the performance characterization in Theorem 1, we consider a server who initiates a federated learning process and selects from a set $\mathcal{K}$ of devices with heterogeneous data to minimize the model accuracy loss. Since the model accuracy loss decreases with the data heterogeneity (i.e., the weighted EMD) of selected devices as shown in Theorem 1, the server may choose and pay rewards to devices with more balanced data (e.g., smaller EMD).

To get the reward from the server, each device $k \in \mathcal{K}$ can cooperate with other devices with complementary data distributions to form a device coalition with more balanced data. We denote the number of device coalitions formed by devices as Z, and denote the coalition formation decision of device k by $h_k \in \mathcal{Z} \triangleq \{1, 2, \ldots, Z\}$ where $h_k = z$ indicates that device k joins device coalition $z \in \mathcal{Z}$. In this case, the number of devices in device coalition z is $l_z = \sum_{k\in\mathcal{K}} \mathbb{1}_{h_k=z}$.

The server selects a subset of device coalitions and each selected device coalition performs model trainings for M communication rounds. Specifically, in each communication round, devices in a selected device coalition perform local trainings based on the current global model, and share model updates with other devices in the same coalition by D2D process, to obtain a consensus model which approximates the centralized learning model trained on the data of all devices in the coalition [12].[4] At the end of the communication round, a representative device uploads the consensus model to the server for global aggregation. Note that in this case, a device coalition acts as a whole to the server, and when calculating the weighted EMD in (4), we should replace the EMD of each selected device k to be the group EMD (GEMD) [25] of each selected device coalition z. We introduce the formal definition of GEMD as follows.

Definition 3 (GEMD): For each device coalition $z \in \mathcal{Z}$, the GEMD measures the divergence between the data distribution of the device coalition and the data distribution of the population dataset, which is

$$\bar{d}_z = \sum_{i=1}^{C} \| \frac{\sum_{k=1}^{l_z} n_k p_k(y=i)}{\sum_{k=1}^{l_z} n_k} - p(y=i)\|. \quad (8)$$

The GEMD quantifies the data heterogeneity of a device coalition. Based on Definition 1 and Definition 3, we can see that $d_k \in [0, 2], \forall k \in \mathcal{K}$, and $\bar{d}_z \in [0, 2], \forall z \in \mathcal{Z}$.

In summary, in coalitional FL, devices with complementary data distributions cooperate and form device coalitions to

[3]Our derived Theorem 1 is based on Proposition 3.1 in [7], and we extend the result $\|\boldsymbol{\omega}_{mT}^{(fed)} - \boldsymbol{\omega}_{mT}^{(cen)}\|$ in Proposition 3.1 in [7] to our novel result $F(\boldsymbol{\omega}_{MT}^{(fed)}) - F(\boldsymbol{\omega}^*)$ in Theorem 1.

[4]To achieve a consensus model, we can let the number of D2D communication rounds in each coalition to be a constant on the order of $O(\log(l^{\max}))$ (according to Proposition 1 in [12]), where $l^{\max}$ is the maximum number of devices in a device coalition characterized later in Proposition 2 in Section V-B1.

maximize their own payoffs, and the server selects an optimal subset of device coalitions to minimize the model accuracy loss. Note that devices' coalition formation behaviors depend on the server's coalition selection strategy, while the server makes its optimal coalition selection decision based on the result of devices' coalition formation interactions. We next model the behaviors of devices and the server, respectively.

B. Devices' Coalition Formation Behaviors

In this subsection, we introduce devices' coalition formation behaviors and model the payoff of each device.

In FL with heterogeneous data, each device $k \in \mathcal{K}$ decides the coalition to join, i.e., $h_k \in \{1, 2, \ldots, Z\}$, to maximize its own payoff. Note that the payoff of a device depends on the server's coalition selection strategy $\boldsymbol{x} = \{x_z \in \{0, 1\} : \forall z \in \mathcal{Z}\}$, where $x_z = 1$ if the server selects device coalition z, and $x_z = 0$ otherwise. If device k joins a device coalition h_k that the server does not select (i.e., $x_{h_k} = 0$), device k will not perform local trainings and cannot receive rewards from the server. In this case, its payoff is 0. If device k joins a device coalition h_k that the server selects (i.e., $x_{h_k} = 1$), device k performs local trainings and shares model updates with other devices in the same device coalition, and finally receives a reward from the server. In this case, device k's payoff includes the reward received from the server, the privacy loss incurred when sharing model updates with other devices in the same device coalition, and the energy cost incurred by performing local trainings.

The reward from the server: Since the model accuracy loss increases with the data heterogeneity of selected device coalitions, a selected device coalition receives a reward that decreases with its GEMD [4]. Without loss of generality, we assume that the reward of device coalition h_k received from the server is[5]

$$\bar{r}_{h_k} = \left(1 - \frac{\bar{d}_{h_k}}{2}\right) r.$$

Here r is the maximum reward the server pays to a device coalition. And we denote $\bar{c}_{h_k} = 1 - \frac{\bar{d}_{h_k}}{2}$ as a score of device coalition h_k. Since $\bar{d}_{h_k} \in [0, 2]$, we have $\bar{c}_{h_k} \in [0, 1]$. Specifically, when device coalition h_k has the maximum GEMD (i.e., $\bar{d}_{h_k} = 2$), its score is $\bar{c}_{h_k} = 0$ and its reward is $\bar{r}_{h_k} = 0$. When device coalition h_k has the minimum GEMD (i.e., $\bar{d}_{h_k} = 0$), its score is $\bar{c}_{h_k} = 1$ and its reward is $\bar{r}_{h_k} = r$.

In a selected device coalition, devices share the reward for fairness, and each device gets a reward which is proportional to its contribution to the device coalition [32]. Similarly, we denote $c_k = 1 - \frac{d_k}{2}$ as a score of device k, which decreases monotonically with its EMD d_k. We calculate the reward that device k can get when it joins device coalition h_k as follows:

$$R_k(h_k, \boldsymbol{h}_{-k}) = \frac{c_k}{\sum_{j:h_j = h_k} c_j} \bar{c}_{h_k} r, \tag{9}$$

where $\boldsymbol{h}_{-k} = \{h_1, h_2, \ldots, h_{k-1}, h_{k+1}, \ldots, h_K\}$ denotes other devices' coalition formation decisions.

The privacy loss due to model sharings: During the model training process, device k performs local trainings and shares its model updates with other devices in the same device coalition. Model sharings in this process would be vulnerable to reconstruction attacks or membership inference attacks from other devices [33]. Thus device k incurs a privacy loss which depends on the number of devices in the same device coalition and device k's data heterogeneity. Specifically, when the number of devices in the device coalition is larger, device k needs to share its model updates with more devices, and hence is more vulnerable to attacks. Furthermore, when device k has more balanced data (or equivalently, a larger score c_k), its data is more valuable to federated learning. Thus the device may suffer more attacks which lead to a larger privacy loss [13], [34]. In summary, we calculate the privacy loss of device k as follows:

$$V_k(h_k, \boldsymbol{h}_{-k}) = (l_{h_k}(h_k, \boldsymbol{h}_{-k}) - 1) c_k \varepsilon, \tag{10}$$

where ε is the privacy sensitivity [34] and $l_{h_k}(h_k, \boldsymbol{h}_{-k}) = \sum_{k' \in \mathcal{K}} \mathbb{1}_{h_{k'} = h_k}$ is the number of devices in device coalition h_k.

The energy cost due to model trainings: When device k performs model trainings, it incurs energy costs including the computation energy cost and the communication energy cost. The computation energy cost of device k is due to local trainings and can be calculated as $E_{comp,k} = \lambda_{comp,k} (f_{comp,k})^2$, where $\lambda_{comp,k}$ is device k's computing coefficient depending on its computing chip and $f_{comp,k}$ is device k's computing frequency (in CPU cycles per second) [35]. The communication energy cost of device k comes from transmitting model updates, and we denote it as $E_{comm,k}$ which depends on device k's communication capacity [30]. The computation energy cost and the communication energy cost do not depend on device k's coalition formation decision. Therefore, we denote the energy cost of device k as E_k, which is a constant in our paper.

In summary, the payoff of device k is

$$P_k(h_k, \boldsymbol{h}_{-k}) = (R_k(h_k, \boldsymbol{h}_{-k}) - V_k(h_k, \boldsymbol{h}_{-k}) - E_k) x_{h_k}. \tag{11}$$

We can see that device k's payoff depends on not only the decision of device k, but also the decisions of other devices. Furthermore, $P_k(h_k, \boldsymbol{h}_{-k})$ changes with h_k and $\boldsymbol{h}_{-k}$ in a highly non-linear manner. This motivates us to analyze devices' behaviors through coalition formation game in Section V.

C. Server's Coalition Selection Problem

In this subsection, we model the server's coalition selection behavior and the optimization problem.

In FL, the server selects a subset $\mathcal{S} \subseteq \mathcal{Z}$ (where $|\mathcal{S}| = S$) of device coalitions for model trainings. In other words, the server's coalition selection strategy $\boldsymbol{x}$ satisfies $\sum_{z=1}^{Z} x_z = S$. Under $\boldsymbol{x}$, the weighted EMD of selected device coalitions is $D_{\mathcal{S}}(\boldsymbol{x}) = \sum_{z:x_z=1} \frac{\bar{n}_z}{\sum_{z':x_{z'}=1} \bar{n}_{z'}} \bar{d}_z$, where $\bar{n}_z = \sum_{k:h_k=z} n_k$ is the number of data samples in device coalition z and $\bar{d}_z$ is the GEMD of device coalition z. Therefore, the weighted EMD

[5]In practice, a device coalition with more balanced data can receive a larger reward from the server [4]. Adopting other decreasing functions will only affect the maximum number of devices in a device coalition, and will not affect our analytical framework and results.

$D_{\mathcal{S}}(\boldsymbol{x})$ depends on the GEMD $\bar{d}_z$ and the number of local data samples $\bar{n}_z$ of selected device coalitions.

The server aims to minimize the model accuracy loss by selecting the optimal subset of device coalitions.[6] As shown in Theorem 1, the model accuracy loss has an upper bound, and we use the upper bound to represent the performance of FL [33], [37] due to the complexity of FL training. Therefore, we can write the server's coalition selection optimization problem as follows:

$$\min_{\boldsymbol{x}} \frac{1}{M\eta(\phi(T-1)(1-\frac{\beta\eta}{2}) - \frac{LG}{\epsilon^2} D_{\mathcal{S}}(\boldsymbol{x}))},$$
$$\text{s.t.} \sum_{z=1}^{Z} x_z = S, x_z \in \{0,1\}, \forall z \in \mathcal{Z}. \tag{12}$$

The objective function depends on both device coalitions' GEMD $\bar{d}_z$ and the number of local data samples $\bar{n}_z$ in a non-monotonous way, which is hard to solve. We will solve the problem in Section VI.

V. Devices' Coalition Formation Game

In this section, we model devices' coalition formation game and analyze its Nash stable partition. Specifically, we will first analyze a special case where the server selects all device coalitions [9] to derive some useful insights, and then analyze the general case where the server selects a subset of device coalitions.

In the following parts, we first define devices' coalition formation game and its Nash stable partition. We then propose an accelerated device coalition formation algorithm to derive the Nash stable partition, which applies to both the special case and the general case. Finally, we characterize the conditions under which the algorithm converges to the Nash stable partition for the two cases.

A. Coalition Formation Game

In FL with heterogeneous data, each device selfishly decides which device coalition to join to maximize its own payoff. We model devices' selfish coalition formation interactions as a coalition formation game. In the following, we first introduce the partition and the preference order of the game. We then formally define the coalition formation game and its Nash stable partition.

First we introduce the concept of partition [38] in coalition formation game. We denote the set of devices in device coalition z as s_z, i.e., $s_z = \{k : h_k = z, k \in \mathcal{K}\}$.[7]

Definition 4 (Partition): A partition Π of device set $\mathcal{K}$ is a set of mutually disjoint device coalitions, i.e., $\Pi = \{s_1, s_2, \ldots, s_Z\}$, where $\bigcup_{z=1}^{Z} s_z = \mathcal{K}$ and $s_z \bigcap s_{z'} = \emptyset, \forall z, z' \in \mathcal{Z}$.

We next introduce the concept of preference order [38] which characterizes a device's preference over any two device coalitions.

Definition 5 (Preference Order): For device $k \in \mathcal{K}$, its preference order over two device coalitions s_z and $s_{z'}$ is $s_z \succeq_k s_{z'}$ (i.e., device k prefers joining device coalition s_z to joining device coalition $s_{z'}$), when (i) the payoff that device k gains by joining device coalition s_z is no smaller than that by joining device coalition $s_{z'}$ (i.e., $P_k(h_k = z, \boldsymbol{h}_{-k}) \geq P_k(h_k = z', \boldsymbol{h}_{-k})$) and (ii) device k's joining does not cause payoff decreases to other devices in device coalition s_z (i.e., $P_j(h_k = z, \boldsymbol{h}_{-k}) \geq P_j(h_k = z', \boldsymbol{h}_{-k}), \forall j \in \mathcal{K}$ and $h_j = z$).

Given the preference order of each device, we model devices' selfish coalition formation interactions in FL as a coalition formation game.

Game 1 (Devices' Coalition Formation Game in FL):
- *Players:* the set $\mathcal{K}$ of devices.
- *Strategies:* each device $k \in \mathcal{K}$ decides the device coalition h_k to join.
- *Payoff functions:* each device $k \in \mathcal{K}$ aims to maximize its payoff $P_k(h_k, \boldsymbol{h}_{-k})$ calculated in (11).
- *Preference orders:* each device $k \in \mathcal{K}$ compares any two device coalitions s_z and $s_{z'}$, and gets the preference order $s_z \succeq_k s_{z'}$ according to Definition 5.

We define the Nash stable partition [39] of Game 1 as follows.

Definition 6 (Nash Stable Partition): A partition $\Pi^* = \{s_1^*, s_2^*, \ldots, s_Z^*\}$ where $s_z^* = \{k : h_k^* = z, k \in \mathcal{K}\}$ is a Nash stable partition of Game 1 if for any device $k \in \mathcal{K}$, we have $s_{h_k^*} \succeq_k s_z \bigcup\{k\}$ for all $s_z \in \Pi^* \bigcup\{\emptyset\}$.

At the Nash stable partition, no device has the incentive to leave its current device coalition to join other device coalitions in the partition. We will analyze devices' coalition formation game in the following subsections.

B. Accelerated Device Coalition Formation Algorithm

Since devices' coalition formation decisions are coupled in a highly non-trivial manner, it's difficult to analyze the existence of the Nash stable partition directly. In this subsection, we propose an accelerated device coalition formation algorithm aiming to derive the Nash stable partition of Game 1. We first introduce the key operations and the acceleration techniques of the algorithm. Then we introduce the main idea and detailed steps of the algorithm. Finally we analyze the complexity of the algorithm.

1) Key Operations and Acceleration Techniques: Our proposed accelerated device coalition formation algorithm involves three key operations, i.e., merge, split, and switch, which mainly depend on the comparison of any two partitions.

To compare two partitions Π and Π', we introduce the concept of Pareto order [39] as follows. We denote $\boldsymbol{h}$ as devices' coalition formation decisions under partition Π, and denote $\boldsymbol{h}'$ as devices' decisions under partition Π'.

Definition 7 (Pareto Order): For any two partitions Π and Π', the Pareto order is $\Pi \triangleright \Pi'$ if $P_k(h_k, \boldsymbol{h}_{-k}) \geq P_k(h'_k, \boldsymbol{h}'_{-k}), \forall k \in \mathcal{K}$ with at least one strict inequality $(>)$ holds.

[6] Here we assume that the server only cares about the performance (i.e., the model accuracy) of FL and has a large enough budget to pay rewards to devices [36].

[7] When there is no confusion, we use device coalition z and device coalition s_z interchangeably.

Intuitively, if $\Pi \triangleright \Pi'$, at least one device gains a larger payoff by forming partition Π than forming partition Π', without decreasing other devices' payoffs. Note that the calculation of the Pareto order depends on the server's coalition selection strategy $\boldsymbol{x}$ as shown in (11).[8]

Based on the Pareto order, we define the operations of merge, split, and switch [23]. Given a partition $\Pi = \{s_1, s_2, \ldots, s_Z\}$, we consider any device coalitions s_z and $s_{z'}$ in Π. We denote s_{-z} as the device coalitions in Π except s_z, and denote $s_{-z,z'}$ as the device coalitions in Π except s_z and $s_{z'}$.

Definition 8 (Coalition Operations): The operations of merge, split, and switch modify partition Π according to the Pareto order as follows.

- *Merge Operation:* If $\{s_z \bigcup s_{z'}, s_{-z,z'}\} \triangleright \{s_z, s_{z'}, s_{-z,z'}\}$, then merge device coalitions s_z and $s_{z'}$ to be one device coalition $s_z \bigcup s_{z'}$.
- *Split Operation:* If $\{s_{z_1}, s_{z_2}, \ldots, s_{z_l}, s_{-z}\} \triangleright \{s_z, s_{-z}\}$ where $\bigcup_{i=1}^{l} s_{z_i} = s_z$, then split device coalition s_z to be device coalitions $s_{z_1}, s_{z_2}, \ldots, s_{z_l}$.
- *Switch Operation:* For a device k in device coalition s_z, if $\{s_z \setminus \{k\}, s_{z'} \bigcup \{k\}, s_{-z,z'}\} \triangleright \{s_z, s_{z'}, s_{-z,z'}\}$, then device k switches from device coalition s_z to device coalition $s_{z'}$.

Intuitively, the above coalition operations happen if devices can get larger payoffs, which depend on the score and the number of devices in the device coalition to join. We next show the necessary conditions regarding the score (Proposition 1) and the number of devices in the device coalition (Proposition 2) that the merge operation and the switch operation satisfy.

Proposition 1: If device coalitions s_z and $s_{z'}$ merge to be one device coalition $s_z \bigcup s_{z'}$ (denoted by z_{merge}), then device coalitions' scores satisfy

$$\bar{c}_{z_{merge}} > \bar{c}_z + \bar{c}_{z'}. \tag{13}$$

If a device k in device coalition s_z switches to device coalition $s_{z'}$ to be $s_{z'} \bigcup \{k\}$ (denoted by z_{switch}), then device coalitions' scores satisfy

$$\bar{c}_{z_{switch}} > \bar{c}_{z'} + \frac{c_k}{\sum_{j \in s_z} c_j} \bar{c}_z. \tag{14}$$

Proof: See Appendix B, available online in the supplementary material. □

Proposition 1 shows that (13) and (14) are the necessary conditions regarding the scores for the merge operation and switch operation to happen. We can use the proposition in the device coalition formation algorithm to eliminate impossible merge or switch operations to accelerate the convergence of the algorithm.

We then characterize the necessary conditions regarding the number of devices in a device coalition. We denote $c_{\min} \triangleq \min_{k \in \mathcal{K}} c_k$ as the minimum score among all devices, and denote $E_{\min} \triangleq \min_{k \in \mathcal{K}} E_k$ as the minimum energy cost among all devices.

[8]We assume that each device can calculate the server's coalition selection strategy $\boldsymbol{x}$ under the current partition Π [40].

Proposition 2: After a merge operation or a switch operation, the number of devices in each device coalition $z \in \mathcal{Z}$, i.e., l_z, has an upper bound. Specifically, when the server selects all device coalitions, each device coalition $z \in \mathcal{Z}$ satisfies

$$l_z \leq \lfloor \frac{\sqrt{(\varepsilon + r)^2 + \frac{4r\varepsilon}{c_{\min}}} - \varepsilon - r}{2\varepsilon} \rfloor. \tag{15}$$

When the server selects a subset of device coalitions, each device coalition $z \in \mathcal{Z}$ satisfies

$$l_z \leq \lfloor \frac{\varepsilon c_{\min} - E_{\min} + \sqrt{(\varepsilon c_{\min} - E_{\min})^2 + 4r\varepsilon c_{\min}}}{2\varepsilon c_{\min}} \rfloor. \tag{16}$$

Proof: See Appendix C, available online in the supplementary material. □

Note that $\lfloor \cdot \rfloor$ is the floor function. Proposition 2 shows that the number of devices in each device coalition has an upper bound $l^{\max}$ which depends on the reward r, the privacy sensitivity ε, $c_{\min}$ and $E_{\min}$. Note that if a coalition operation generates a device coalition whose number of devices is larger than $l^{\max}$, devices have no incentive to perform the operation. Thus we can use Proposition 2 to eliminate impossible merge or switch operations to accelerate the coalition formation process.

2) Main Idea and Detailed Steps of the Algorithm: With the key operations and acceleration techniques, we introduce the accelerated device coalition formation algorithm (Algorithm 1) to derive the Nash stable partition of Game 1.

Main Idea: The coalition formation is a process where considering the server's coalition selection strategy, devices iteratively decide the device coalitions to join. The algorithm performs possible merge, split, and switch operations repeatedly until reaching the Nash stable partition. Specifically, the initial partition is $\Pi_{initial} = \{s_1, s_2, \ldots, s_K\}$ where $s_k = \{k\}, \forall k \in \mathcal{K}$, i.e., each device forms a device coalition by itself. Each device $k \in \mathcal{K}$ first calculates its EMD d_k (Line 1). Then devices perform the accelerated merge, split, and accelerated switch operations in turn repeatedly (Lines 3-8). The iteration stops when no device performs any operation under the current partition and the algorithm converge to the Nash stable partition.

Accelerated Merge Operations: We show the accelerated merge operations in Algorithm 2. Specifically, if a merge operation of two device coalitions in the current partition satisfies Propositions 1 and 2 (Line 3), and the new partition after the merge operation satisfies the Pareto order over the current partition (Line 4), then merge the two device coalitions and update the current partition (Lines 5-6). We repeat the process for all device coalitions in the current partition. Note that compared with traditional coalition formation algorithms that check the Pareto order for the merge of every two device coalitions exhaustively, we can eliminate the impossible merge operations by using the acceleration techniques in Propositions 1 and 2 to accelerate the process.

Split Operations: We show the split operations in Algorithm 3. Specifically, for $s_z \in \Pi_{updated}$, we first generate the possible split set $\mathcal{S}_z^{split}$ (Line 2), which consists of all possible splitting

Algorithm 1: Accelerated Device Coalition Formation Algorithm.

Input: The initial partition $\Pi_{initial} = \{s_1, s_2, \cdots, s_K\}$
Output: The final partition Π^*
Each device $k \in \mathcal{K}$ calculates its EMD d_k;
Set $\Pi_{updated} \leftarrow \Pi_{initial}$ and $\Pi_{prev} \leftarrow \emptyset$;
while $\Pi_{updated} \neq \Pi_{prev}$ **do**
  Set $\Pi_{prev} \leftarrow \Pi_{updated}$;
  Perform the **accelerated merge operations** in Algorithm 2 and set $\Pi_{updated} \leftarrow \Pi_{merge}$;
  Perform the **split operations** in Algorithm 3 and set $\Pi_{updated} \leftarrow \Pi_{split}$;
  Perform the **accelerated switch operations** in Algorithm 4 and set $\Pi_{updated} \leftarrow \Pi_{switch}$;
end
Set $\Pi^* \leftarrow \Pi_{updated}$.

Algorithm 2: Accelerated Merge Algorithm.

Input: The current partition $\Pi_{updated}$, the EMD $d_k, \forall k \in \mathcal{K}$
Output: The partition Π_{merge} after merge operations
for *each device coalition* $s_z \in \Pi_{updated}$ **do**
  Consider a hypothetical merge operation with device coalition $s_{z'}$ and set $s_{new} \leftarrow s_z \bigcup s_{z'}$;
  if *the operation satisfies Propositions 1 and 2* **then**
    if $\{s_{new}, s_{-z,z'}\} \triangleright \{s_z, s_{z'}, s_{-z,z'}\}$ **then**
      Merge s_z and $s_{z'}$ to be s_{new};
      Set $\Pi_{updated} \leftarrow \{s_{new}, s_{-z,z'}\}$;
    end
  end
end
Set $\Pi_{merge} \leftarrow \Pi_{updated}$.

Algorithm 3: Split Algorithm.

Input: The current partition $\Pi_{updated}$, the EMD $d_k, \forall k \in \mathcal{K}$
Output: The partition Π_{split} after split operations
1 **for** *each* $s_z \in \Pi_{updated}$ *satisfying* $l_z > 1$ **do**
2   Generate the possible split set of s_z as $\mathcal{S}_z^{split}$;
3   **for** $\{s_{z_1}, s_{z_2}, \cdots, s_{z_j}\} \in \mathcal{S}_z^{split}$ **do**
4   **if** $\{s_{z_1}, s_{z_2}, \cdots, s_{z_j}, s_{-z}\} \triangleright \{s_z, s_{-z}\}$ **then**
5   Split s_z to be $\{s_{z_1}, s_{z_2}, \cdots, s_{z_j}\}$;
6   Set $\Pi_{updated} \leftarrow \{s_{z_1}, \cdots, s_{z_j}, s_{-z}\}$;
7   Break;
8   **end**
9   **end**
10 **end**
11 Set $\Pi_{split} \leftarrow \Pi_{updated}$.

results of s_z. Then, we find a possible splitting result which satisfies the Pareto order and perform the corresponding split operation. Specifically, if a possible splitting result $\{s_{z_1}, s_{z_2}, \ldots, s_{z_j}\}$ of s_z brings a new partition $\{s_{z_1}, \ldots, s_{z_j}, s_{-z}\}$ that satisfies the Pareto order over the current partition $\{s_z, s_{-z}\}$, we split coalition s_z into $\{s_{z_1}, \ldots, s_{z_j}\}$ (Line 5), update the partition with $\Pi_{updated} \leftarrow \{s_{z_1}, \ldots, s_{z_j}, s_{-z}\}$ (Line 6), and skip checking other splitting results in set $\mathcal{S}_z^{split}$ (Line 7). We repeat the process for all device coalitions in the current partition. Note that since the maximum number of devices in each device coalition

Algorithm 4: Accelerated Switch Algorithm

Input: The current partition $\Pi_{updated}$, the EMD $d_k, \forall k \in \mathcal{K}$
Output: The partition Π_{switch} after switch operations
1 **for** *each* $s_z \in \Pi_{updated}$ *satisfying* $l_z > 1$ **do**
2   **for** *each device* k *in device coalition* s_z **do**
3   Consider a hypothetical switch operation to join device coalition $s_{z'}$;
4   **if** *the operation satisfies Propositions 1 and 2* **then**
5   **if** $\{s_z \setminus \{k\}, s_{z'} \bigcup \{k\}, s_{-z,z'}\} \triangleright \{s_z, s_{z'}, s_{-z,z'}\}$ **then**
6   Device k switches from s_z to $s_{z'}$;
7   Set $\Pi_{updated} \leftarrow \{s_z \setminus \{k\}, s_{z'} \bigcup \{k\}, s_{-z,z'}\}$;
8   **end**
9   **end**
10   **end**
11 **end**
12 Set $\Pi_{switch} \leftarrow \Pi_{updated}$.

has an upper bound as shown in Proposition 2, the number of possible split operations is not very large.

Accelerated Switch Operations: We show the accelerated switch operations in Algorithm 4. Specifically, if a switch operation of a device from its current device coalition to another device coalition satisfies Propositions 1 and 2 (Line 4), and the new partition after the switch operation satisfies the Pareto order over the current partition (Line 5), then switch the device to the new device coalition and update the current partition (Lines 6-7). We repeat the process for all devices in the current partition. Similarly, compared with traditional coalition formation algorithms that check the Pareto order for every switch operation exhaustively, we can accelerate the algorithm by using the two acceleration techniques in Propositions 1 and 2.

3) Complexity Analysis: In this part, we analyze the complexity of our proposed algorithm. We first analyze the complexities of Algorithms 2, 3, and 4. Then we discuss the complexity of Algorithm 1.

We first analyze the complexity of the accelerated merge, split, and accelerated switch algorithms.

Lemma 1: The complexity of the accelerated merge and accelerated switch algorithms (i.e., Algorithms 2 and 4) is $O(K^3)$, and the complexity of the split algorithm (i.e., Algorithm 3) is $O(K2^{l^{\max}})$.

Proof: See Appendix D, available online in the supplementary material. □

Note that our two acceleration techniques in Propositions 1 and 2 help avoid many impossible merge and switch operations. For the split algorithm, the complexity depends on the maximum number of devices in a device coalition, which is usually not large as we find in simulations.

The accelerated device coalition formation algorithm (Algorithm 1) performs the accelerated merge, split, and accelerated switch algorithms iteratively. In each iteration, the complexity is $O(\max\{K^3, K2^{l^{\max}}\})$. It is challenging to quantify the number of iterations for the coalition formation process to converge to a Nash stable partition [23], since it depends on the number of devices, the initial partition, and both the data heterogeneity and the number of data samples of devices. Our simulation results show that even when the number of devices is large, Algorithm 1 converges to the final partition in a few iterations. We next characterize the conditions under which Algorithm 1 converges to a Nash stable partition of Game 1.

C. Analysis of Nash Stable Partition When Server Selects all Device Coalitions

In this subsection, we analyze the special case where the server selects all device coalitions [9] and analyze the Nash stable partition of devices' coalition formation game (Game 1) in this case. In the following, we first define a loop of partitions which may appear in the coalition formation process. Then we show that the loop does not exist in the special case. Finally we prove that the Nash stable partition Π^* always exists in the special case and Algorithm 1 converges to Π^*.

In the coalition formation process in Algorithm 1, the partition of devices changes due to the merge, split, and switch operations. In this process, a loop [41] of partitions may appear as follows.

Definition 9 (Partition Loop): A loop is a sequence of partitions $(\Pi_1, \Pi_2, \ldots, \Pi_W, \Pi_{W+1} = \Pi_1)$, where at least one coalition operation of merge, split, or switch changes partition Π_w to be partition Π_{w+1}, and $\Pi_w \neq \Pi_{w'}$ for all $1 \leq w < w' \leq W, W \geq 3$.

If a loop of partitions exists in the coalition formation process in Algorithm 1, the algorithm may fall into the loop and never converge to the Nash stable partition Π^*. We next prove that when the server selects all device coalitions, the loop of partitions does not exist.

Lemma 2: When the server selects all device coalitions, any partition will appear at most once in the coalition formation process in Algorithm 1, and the partition loop does not exist in this case.

Proof: See Appendix E, available online in the supplementary material. □

Lemma 2 shows that when the server selects all device coalitions, the coalition formation process in Algorithm 1 will never fall in a loop. In this case, Algorithm 1 will always converge since the number of possible device partitions is finite. Next we show that Algorithm 1 converges to the Nash stable partition.

Theorem 2: When the server selects all device coalitions, the Nash stable partition Π^* of Game 1 always exists and Algorithm 1 converges to the Nash stable partition Π^*.

Proof: See Appendix F, available online in the supplementary material. □

Theorem 2 shows that when the server selects all device coalitions, devices can reach the Nash stable partition after a finite number of coalition operations (i.e., merge, split, or switch) in Algorithm 1.

D. Analysis of Nash Stable Partition When Server Selects a Subset of Device Coalitions

In this subsection, we focus on the general case where the server selects a subset of device coalitions and analyze the Nash stable partition of devices' coalition formation game in this case. We first prove that the Nash stable partition may not always exist. Then we characterize the condition for the existence of the Nash stable partition.

We first show that the Nash stable partition does not always exist in the general case.

Theorem 3: When the server selects a subset of device coalitions, the Nash stable partition of Game 1 may not exist.

Proof: See Appendix G, available online in the supplementary material. □

Our analysis shows that the existence of the Nash stable partition depends on devices' privacy sensitivity ε. We next characterize the condition under which the Nash stable partition of Game 1 exists.

Theorem 4: When the server selects a subset of device coalitions, there exists a threshold ε_{thres} of devices' privacy sensitivity. When $\varepsilon > \varepsilon_{thres}$, the Nash stable partition Π^* of Game 1 always exists and Algorithm 1 converges to Π^*. Especially, when $\varepsilon > \frac{r-E_{\min}}{c_{\min}}$, each device forms a device coalition by itself at the Nash stable partition. When $\varepsilon \leq \varepsilon_{thres}$, the Nash stable partition Π^* of Game 1 may not exist.

Proof: See Appendix H, available online in the supplementary material. □

Theorem 4 shows that when the privacy sensitivity is large, i.e., $\varepsilon > \varepsilon_{thres}$, there is no partition loop in the coalition formation process, and devices can reach the Nash stable partition in Algorithm 1. Especially, when the privacy sensitivity is vary large, i.e., $\varepsilon > \frac{r-E_{\min}}{c_{\min}}$, forming a coalition with other devices causes a high privacy loss, and each device forms a device coalition by itself. When the privacy sensitivity is small, i.e., $\varepsilon \leq \varepsilon_{thres}$, the privacy loss when forming coalitions with other devices is small and devices are willing to perform many possible coalition operations, which may lead to a partition loop.

Based on the analysis of devices' coalition formation behaviors, we next analyze the server's coalition selection problem.

VI. Server's Optimal Coalition Selection

In this section, we analyze the server's coalition selection problem. We first analyze the properties of the problem. Then we propose a low-complexity algorithm to derive the optimal coalition selection strategy for the server.

A. Server's Coalition Selection Optimization Problem

In this subsection, we present the server's coalition selection optimization problem and analyze its properties.

As discussed in Section IV-C, the server selects an optimal subset of device coalitions to minimize the model accuracy loss, which is

$$\min_{\boldsymbol{x}} \frac{1}{M\eta(\phi(T-1)(1-\frac{\beta\eta}{2}) - \frac{LG}{\epsilon^2} D_{\mathcal{S}}(\boldsymbol{x}))}$$
$$\text{s.t.} \sum_{z=1}^{Z} x_z = S, x_z \in \{0,1\}, \forall z \in \mathcal{Z}.$$

Since the objective function increases with the weighted EMD $D_{\mathcal{S}}(\boldsymbol{x})$, the optimization problem is equivalent to

$$\min_{\boldsymbol{x}} \; D_{\mathcal{S}}(\boldsymbol{x}) = \sum_{z:x_z=1} \frac{\bar{n}_z}{\sum_{z':x_{z'}=1} \bar{n}_{z'}} \bar{d}_z$$

$$\text{s.t.} \; \sum_{z=1}^{Z} x_z = S, x_z \in \{0,1\}, \forall z \in \mathcal{Z}. \tag{17}$$

We can see that the weighted EMD $D_{\mathcal{S}}(\boldsymbol{x})$ increases with selected device coalitions' GEMD $\bar{d}_z$, while changes with the number of data samples $\bar{n}_z$ in a non-monotonous way, for all $z \in \{z : x_z = 1, z \in \mathcal{Z}\}$. Therefore, $D_{\mathcal{S}}(\boldsymbol{x})$ depends on the coalition selection strategy $\boldsymbol{x}$ in a complex way. Note that the number of data samples $\bar{n}_z$ in a device coalition depends on devices' coalition formation behaviors, i.e., $\bar{n}_z = \sum_{k:h_k=z} n_k$, while devices' coalition formation behaviors at the Nash stable partition depends on the server's coalition selection strategy non-explicitly. Therefore, it is challenging to derive the closed-form solution to the server's coalition selection optimization problem.

B. Optimal Coalition Selection Algorithm

In this subsection, we propose a low-complexity algorithm to derive the optimal coalition selection strategy for the server. We first introduce the main idea of the algorithm, which breaks the coalition selection optimization problem into two subproblems. We then introduce the solution method to each subproblem. We finally discuss the optimality and complexity performance of the algorithm.

The main idea of our proposed optimal coalition selection algorithm is to break the coalition selection problem (17) into two subproblems. Specifically, the first subproblem selects the optimal subset of device coalitions to minimize $D_{\mathcal{S}}(\boldsymbol{x})$, under a constraint $D_{\mathcal{S}}(\boldsymbol{x}) \le u$ for a fixed value u, which is

$$\min_{\boldsymbol{x}} \; \frac{\sum_{z=1}^{Z} \bar{n}_z \bar{d}_z x_z}{\sum_{z=1}^{Z} \bar{n}_z x_z} \tag{18a}$$

$$\text{s.t.} \; \frac{\sum_{z=1}^{Z} \bar{n}_z \bar{d}_z x_z}{\sum_{z=1}^{Z} \bar{n}_z x_z} \le u, \tag{18b}$$

$$\sum\nolimits_{z=1}^{Z} x_z = S, x_z \in \{0,1\}, \forall z \in \mathcal{Z}. \tag{18c}$$

The second subproblem finds the minimum value of u under which there is a feasible coalition selection strategy to problem (18).

We first solve problem (18) under a fixed value u. Note that since $\bar{n}_z > 0, \forall z \in \mathcal{Z}$, constraint (18b) is equivalent to

$$\sum_{z=1}^{Z} x_z \bar{n}_z (\bar{d}_z - u) \le 0. \tag{19}$$

To find a subset of device coalitions satisfying (19), we sort device coalitions in an ascending order of $\bar{n}_z(\bar{d}_z - u)$ and select the first S device coalitions. If such a coalition selection strategy satisfies (19), then the optimal objective value of problem (18) is no larger than u. Note that in this process, we use the metric $\bar{n}_z(\bar{d}_z - u)$ to capture the comprehensive impact of both $\bar{d}_z$ and $\bar{n}_z$.

Algorithm 5: Optimal Coalition Selection Algorithm.

Input: The number of samples $\bar{n}_z$ and the GEMD $\bar{d}_z$ of each device coalition, the partition Π
Output: The optimal coalition selection strategy $\boldsymbol{x}^*$
1 Set $u_l = 0, u_r = 2$;
2 **while** $u_r - u_l > \varrho$ **do**
3 Set $u \leftarrow (u_l + u_r)/2$;
4 The server announces u to device coalitions ;
5 Each device coalition calculates $\bar{n}_z(\bar{d}_z - u)$ and reports to the server;
6 Sort device coalitions in an ascending order of $\bar{n}_z(\bar{d}_z - u)$ and select the first S device coalitions $\mathcal{U} \leftarrow \{s_1, s_2, \cdots, s_S\}$;
7 Calculate $U \leftarrow \sum_{i=1}^{S} \bar{n}_z(\bar{d}_z - u)$;
8 **if** $U \le 0$ **then**
9 $u_r \leftarrow u$;
10 **else**
11 $u_l \leftarrow u$;
12 **end**
13 **end**
14 Set $\boldsymbol{x}^* = \{x_z^* : x_z^* = 1, \forall s_z \in \mathcal{U}, x_z^* = 0, \forall s_z \notin \mathcal{U}\}$.

Then we solve the second subproblem which finds the minimum value of u, denoted by $u_{\min}$, under which there is a feasible coalition selection strategy to problem (18). Given $u_{\min}$, we sort device coalitions in an ascending order of $\bar{n}_z(\bar{d}_z - u_{\min})$ and select the first S device coalitions. If such a coalition selection strategy satisfies (19), then the coalition selection strategy is the optimal solution to problem (17) and $u_{\min}$ is the optimal objective value. Note that since $\bar{d}_z \in [0,2], \forall z \in \mathcal{Z}$ as discussed in Section IV-A, we have $u \in [0,2]$. To find the value $u_{\min}$, we use the dichotomy method for $u \in [0,2]$ with an error tolerance level ϱ (Line 2 in Algorithm 5).

We summarize our proposed optimal coalition selection algorithm in Algorithm 5. We show the performance of Algorithm 5 in the following lemma.

Lemma 3: Algorithm 5 derives the optimal coalition selection strategy $\boldsymbol{x}^*$ to problem (17) with complexity $O(K \log \frac{2K}{\varrho})$.

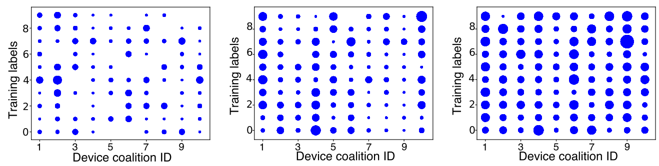
Proof: See Appendix I, available online in the supplementary material. □

Lemma 3 shows that Algorithm 5 has a low complexity and it can guide the server to select the optimal subset of device coalitions effectively.

VII. Simulation Results

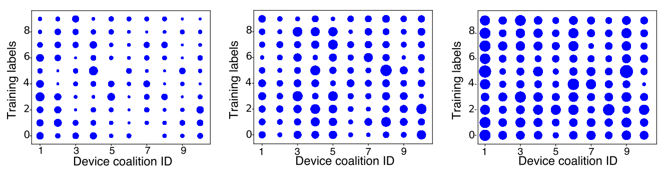
In this section, we conduct numerical experiments to evaluate the performance of our proposed coalition formation and selection framework. We first compare our method with a baseline method and several methods in existing works. We then evaluate the impact of several system parameters.

We consider a FL system with $K = 100$ devices on the dataset MNIST which contains 60000 figures of handwritten digits. We assume that the server selects $S = 10$ devices or device coalitions to perform model trainings. We model devices' heterogeneous data distributions using a Dirichlet distribution



(a) Baseline (b) $r = 10$ (c) $r = 20$

Fig. 2. Data distributions of selected device coalitions when $\alpha = 0.4$.



(a) Baseline (b) $r = 10$ (c) $r = 20$

Fig. 3. Data distributions of selected device coalitions when $\alpha = 0.8$.

Dir(α) on MNIST, where a smaller α indicates a higher data heterogeneity [42]. We assume that the FL process between the server and devices involves $M = 300$ communication rounds where devices perform $T = 5$ steps of local trainings in each communication round, and the learning rate is $\eta = 0.01$ [42]. We assume that the reward is $r = 40$, the privacy sensitivity is $\varepsilon = 2$, and the energy cost of each device is $E = 1$.

A. Comparisons With Existing Methods

In this subsection, we compare our proposed coalition formation and selection framework with several methods in existing works. We first introduce a baseline method (where each device forms a coalition by itself) and three related methods in existing works. We then show the data distributions and the GEMD of selected device coalitions under the baseline method and under our proposed coalition formation and selection framework. Then we compare the performance of our method with the performances of four existing methods.

We first introduce a baseline method and three related methods which target on FL with heterogeneous data [7], [17], [18].

- *Baseline:* each device forms a device coalition by itself.
- *Data sharing [7]*: the server shares a small subset of global data to devices.
- *Fed-ZDAC [17]*: the server generates fake data using the zero-shot data generation method and mixes the real local data and the fake data.
- *FAuG [18]*: devices upload few data samples to the server to train a GAN to generate data samples.

To show the effectiveness of our proposed coalition formation and selection framework on tackling the data heterogeneity problem in FL, we show the data distributions (in Figs. 2 and 3) and the GEMD (in Figs. 4 and 5) of the selected $S = 10$ device coalitions under the baseline method and under our proposed framework (for $r = 10$ and $r = 20$). Specifically, Figs. 2 and 3 show the data distributions of the selected device coalitions when $\alpha = 0.4$ and $\alpha = 0.8$ respectively, where a smaller α indicates a higher data heterogeneity among devices. In each sub-figure, the size of a scattered point indicates the number of training samples for a label in a device coalition. Both Fig. Figs. 2 and 3 show that under the baseline method, selected device coalitions have unbalanced data which lacks some labels and the number of data samples of different labels is biased. Under our proposed framework, selected device coalitions have more balanced data for both $r = 10$ and $r = 20$, compared with that under the baseline method, and the number of data samples for each label and each selected device coalition increases significantly when $r = 20$ in both Figs. 2 and 3. This validates the effectiveness of our proposed framework under different levels of data heterogeneity.

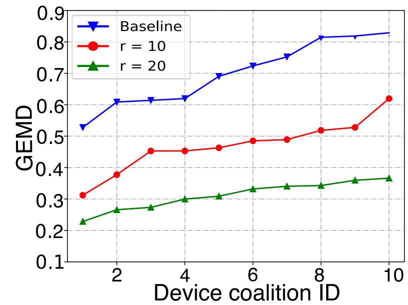


Fig. 4. GEMD of selected device coalitions when $\alpha = 0.4$.

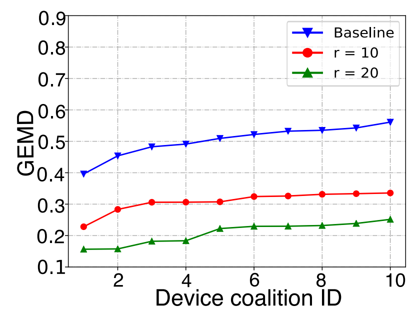


Fig. 5. GEMD of selected device coalitions when $\alpha = 0.8$.

Figs. 4 and 5 show the GEMD of selected device coalitions for the cases in Figs. 2 and 3 respectively. We can see that the GEMD of each selected device coalition is smaller under our proposed coalition formation and selection framework than that under the baseline method. Furthermore, we can calculate that the weighted EMD of selected device coalitions under our proposed framework is reduced by up to 58.6% than that under the baseline method, which indicates that our proposed coalition formation and selection framework can effectively solve the data heterogeneity problem in FL. This is because under our proposed framework, devices with complementary data distributions cooperate to form device coalitions with more balanced data.

We then compare the model accuracy achieved by our proposed method and the ones achieved by the four related methods (i.e., the baseline method and methods in [7], [17], [18]) in Fig. 6. We can see that our method achieves a higher model accuracy compared with the four related methods in all communication rounds. When the training process converges, the model accuracy of our method is 87.9%, while the highest model accuracy of

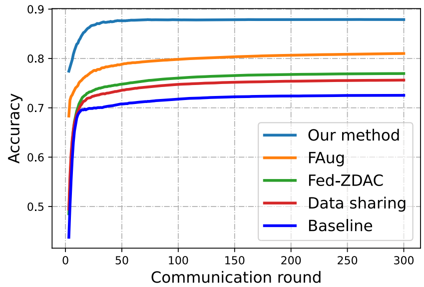


Fig. 6. Comparison with four methods.

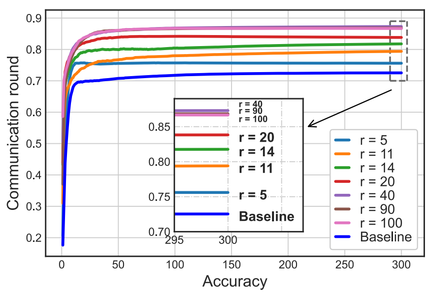


Fig. 7. Accuracy under different r.

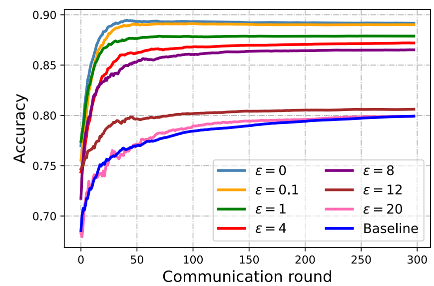


Fig. 8. Accuracy under different ε.

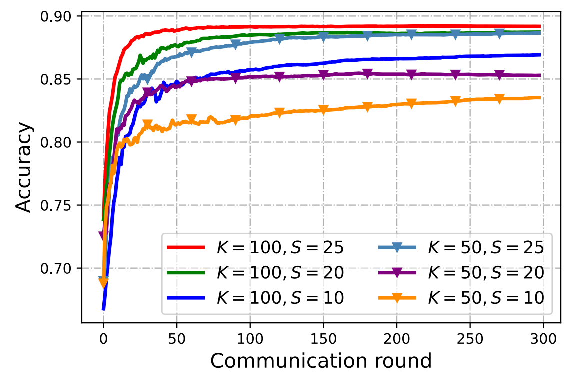


Fig. 9. Accuracy under different K and S.

the four related methods is 81.1%. Hence, our method improves the model accuracy by 6.8%. The performance improvement is due to two reasons. First, devices with complementary data distributions cooperate and form device coalitions with more balanced data. Second, the server selects the optimal subset of device coalitions to minimize the model accuracy loss effectively. Thus our proposed method achieves a better model accuracy.

B. Impact of System Parameters

In this subsection, we evaluate the impact of system parameters on the performance of our proposed method. Specifically, we show how the model accuracy of FL changes with the reward r, the privacy sensitivity ε, and the total number of devices K as well as the number of selected device coalitions S. We also conduct simulations to show the impact of data heterogeneity on model accuracy as well as the accuracy improvement brought by our proposed coalitional FL, and we put the simulation results in Appendix J, available online due to page limit.

The impact of the reward r: Fig. 7 shows the model accuracy achieved by FL under different values of r. We can see that when the training process converges, the model accuracy achieved by the FL process first increases with r when $r \leq 40$, and then remains unchanged when $r > 40$. Specifically, the model accuracy increases with r when $r \leq 40$. The reason is that when $r \leq 40$, as r increases, devices can get more rewards by joining a device coalition and hence may form device coalitions with more balanced data. In this case, the weighted EMD of selected device coalitions is smaller and our method can achieve a better model accuracy. When $r > 40$, the model accuracy remains unchanged. The reason is that when r is large enough, devices already form device coalitions with balanced data, and a larger r cannot incentive devices to form device coalitions with more balanced data. Thus the model accuracy remains unchanged.

The impact of the privacy sensitivity ε: Fig. 8 shows the model accuracy achieved by FL under different values of ε. We can see that when the training process converges, the model accuracy decreases with the privacy sensitivity ε. The reason is that when the privacy sensitivity increases, devices will incur larger privacy losses when joining a device coalition, especially when joining a device coalition with many devices. In this case, each device coalition at the Nash stable partition will have a smaller number of devices, which is consistent with our analysis in Proposition 2 in Section V-B. Thus the data distribution of selected device coalitions may be more biased under a larger ε, which leads to a lower model accuracy. When the privacy sensitivity $\varepsilon \geq 20$, no device will form device coalitions with other devices at the Nash stable partition, and the model accuracy achieved by FL will not change with ε.

The impact of the total number of devices K as well as the number of selected device coalitions S: Fig. 9 shows the model accuracy achieved by FL under different values of K and S. We can see that when the training process converges, the model accuracy increases with K and S. Specifically, given a fixed number of devices K, the model accuracy increases with the number of selected device coalitions S. The reason is that selecting more device coalitions can effectively prevent overfitting and improve the model accuracy. Similarly, given a fixed number of selected device coalitions S, the model accuracy increases with the total number of devices K. The reason is that when K is large, devices can form more possible device coalitions. Thus the data distribution of the selected device coalitions may be more balanced, which can improve the model accuracy.

VIII. Conclusion

In this work, we propose a novel coalitional FL framework for FL with heterogeneous data where devices with complementary data distributions cooperate to form device coalitions and the server selects the optimal subset of device coalitions for model trainings. We first derive a novel theoretical relationship between data heterogeneity and model accuracy, based on which we analyze devices' formation game. We characterize the conditions under which the Nash stable partition exists and propose an accelerated algorithm for devices to reach the Nash stable partition. We then propose a low-complexity algorithm for the server to select the optimal subset of device coalitions efficiently. Simulation results show that our coalition formation and selection framework can reduce the data heterogeneity of selected device coalitions by up to 58.6%, and increase the model accuracy by up to 6.8% compared with four benchmarks. There are many interesting directions to study in the future. For example, it is of practical importance to consider the incomplete information scenario where devices do not know the coalition formation strategies of other devices or the data heterogeneity of other devices. It is also interesting to design proper reward mechanisms to selected device coalitions.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2350–2358.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[4] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1035–1044, Second Quarter, 2022.

[5] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1739–1748.

[6] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, *arXiv: 1910.14425*.

[7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv: 1806.00582*.

[8] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Jan. 2021.

[9] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1, pp. 1–210, 2021.

[10] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[11] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[12] S. Hosseinalipour et al., "Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1569–1584, Aug. 2022.

[13] C. Huang, H. Zhang, and X. Liu, "Incentivizing data contribution in cross-silo federated learning," 2022, *arXiv:2203.03885*.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, no. 1, pp. 429–450, 2020.

[15] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," 2021, *arXiv:2101.11203*.

[16] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, *arXiv: 2002.07948*.

[17] W. Hao et al., "Towards fair federated learning with zero-shot data augmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3310–3319.

[18] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv: 1811.11479*.

[19] Y. Wang, J. Wolfrath, N. Sreekumar, D. Kumar, and A. Chandra, "Accelerated training via device similarity in federated learning," in *Proc. 4th Int. Workshop Edge Syst. Analytics Netw.*, 2021, pp. 31–36.

[20] R. Lu et al., "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1145–1158, Apr. 2023.

[21] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: Clients clustering for better personalization," *World Wide Web*, vol. 26, no. 1, pp. 481–500, 2023.

[22] S. Arisdakessian, O. A. Wahab, A. Mourad, and H. Otrok, "Coalitional federated learning: Improving communication and training on non-IID data with selfish clients," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2462–2476, Jul./Aug. 2023.

[23] J. S. Ng et al., "Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2675–2686, Nov. 2022.

[24] L. Wu, S. Guo, Y. Ding, Y. Zhan, and J. Zhang, "A coalition formation game approach for personalized federated learning," 2022, *arXiv:2202.02502*.

[25] J. Ma, X. Sun, W. Xia, X. Wang, X. Chen, and H. Zhu, "Client selection based on label quantity information for federated learning," in *Proc. IEEE Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2021, pp. 1–6.

[26] W. Guo, R. Li, C. Huang, X. Qin, K. Shen, and W. Zhang, "Joint device selection and power control for wireless federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2395–2410, Aug. 2022.

[27] S. R. Pandey, L. D. Nguyen, and P. Popovski, "A contribution-based device selection scheme in federated learning," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2057–2061, Sep. 2022.

[28] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16 478–16 488.

[29] X. Fang and M. Ye, "Robust federated learning with noisy and heterogeneous clients," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 072–10 081.

[30] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3910–3924, Jul. 2023.

[31] C. Chen, Z. Chen, Y. Zhou, and B. Kailkhura, "FedCluster: Boosting the convergence of federated learning via cluster-cycling," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 5017–5026.

[32] J. Zhang, C. Li, A. Robles-Kelly, and M. Kankanhalli, "Hierarchically fair federated learning," 2020, *arXiv: 2004.10386*.

[33] P. Sun et al., "Pain-FL: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Dec. 2021.

[34] G. Liao, X. Chen, and J. Huang, "Privacy policy in online social network with targeted advertising business," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 934–943.

[35] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[36] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[37] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[38] J. S. Ng et al., "Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2326–2344, Apr. 2021.

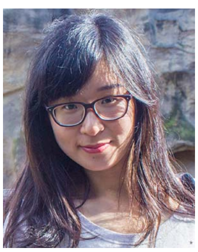
[39] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 77–97, Sep. 2009.

[40] R. Hu and Y. Gong, "Trading data for learning: Incentive mechanism for on-device federated learning," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.

[41] Q. Ma, E. Yeh, and J. Huang, "How bad is selfish caching?," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2019, pp. 11–20.

[42] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12 878–12 889.

Ning Zhang received the BS and MS degrees from the School of Intelligent Systems Engineering, Sun Yat-sen University (SYSU), Shenzhen, China in 2020 and 2023, respectively. He is currently a researcher with Research Institute of China Telecom Company Ltd., China. His primary research interests include network design, network optimization, incentive mechanism and federated learning.



Qian Ma (Member, IEEE) received the BS degree from the Beijing University of Posts and Telecommunications (China) in 2012 and the PhD degree from the Department of Information Engineering, the Chinese University of Hong Kong in 2017. She is an associate professor with the School of Intelligent Systems Engineering, Sun Yat-sen University. She worked as a postdoc research associate with Northeastern University during 2018-2019. Her research interests lie in the field of network optimization and economics. She is the recipient of the Best Paper Award from the IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) in 2021 and the recipient of the Best Student Paper Award from the IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) in 2015.

Wuxing Mao received the BE degree from the School of Intelligent Systems Engineering, Sun Yat-sen University (SYSU), Shenzhen, China in 2021. He is currently working toward the MS degree with the School of Intelligent Systems Engineering, SYSU. His primary research interests include incentive mechanism, data privacy, and federated learning.

Xu Chen received the PhD degree in information engineering from the Chinese University of Hong Kong in 2012. He is a full professor with Sun Yat-sen University, Guangzhou, China, and the vice director of National and Local Joint Engineering Laboratory. He worked as a postdoctoral research associate with Arizona State University, Tempe, USA, from 2012 to 2014, and a Humboldt Scholar fellow with the Institute of Computer Science of the University of Goettingen, Germany from 2014 to 2016. He received the prestigious Humboldt research fellowship awarded by the Alexander von Humboldt Foundation of Germany, 2014 Hong Kong Young Scientist Runner-up Award, 2017 IEEE Communication Society Asia-Pacific Outstanding Young Researcher Award, 2017 IEEE ComSoc Young Professional Best Paper Award, Honorable Mention Award of 2010 IEEE international conference on Intelligence and Security Informatics (ISI), Best Paper Runner-up Award of 2014 IEEE International Conference on Computer Communications (INFOCOM), and Best Paper Award of 2017 IEEE International Conference on Communications (ICC). He is currently an area editor of the IEEE Open Journal of the Communications Society, an associate editor of the *IEEE Transactions Wireless Communications*, *IEEE Transactions Vehicular Technology*, and *IEEE Internet of Things Journal*.