# Social-Aware Clustered Federated Learning With Customized Privacy Preservation

Yuntao Wang, Zhou Su, *Senior Member, IEEE*, Yanghe Pan, Tom H. Luan, *Senior Member, IEEE*, Ruidong Li, *Senior Member, IEEE*, and Shui Yu, *Fellow, IEEE*

***Abstract*— A key feature of federated learning (FL) is to preserve the data privacy of end users. However, there still exist potential privacy leakage in exchanging gradients under FL. As a result, recent research often explores the differential privacy (DP) approaches to add noises to the computing results to address privacy concerns with low overheads, which however degrade the model performance. In this paper, we strike the balance of data privacy and efficiency by utilizing the pervasive social connections between users. Specifically, we propose SCFL, a novel Social-aware Clustered Federated Learning scheme, where mutually trusted individuals can freely form a social cluster and aggregate their raw model updates (e.g., gradients) inside each cluster before uploading to the cloud for global aggregation. By mixing model updates in a social group, adversaries can only eavesdrop the social-layer combined results, but not the privacy of individuals. As such, SCFL considerably enhances model utility without sacrificing privacy in a low-cost and highly feasible manner. We unfold the design of SCFL in three steps. *i) Stable social cluster formation*. Considering users' heterogeneous training samples and data distributions, we formulate the optimal social cluster formation problem as a federation game and devise a fair revenue allocation mechanism to resist free-riders. *ii) Differentiated trust-privacy mapping*. For the clusters with low mutual trust, we design a customizable privacy preservation mechanism to adaptively sanitize participants' model updates depending on social trust degrees. *iii) Distributed convergence*. A distributed two-sided matching algorithm is devised to attain an optimized disjoint partition with Nash-stable convergence. Experiments on Facebook network and MNIST/CIFAR-10 datasets validate that our SCFL can effectively enhance learning utility, improve user payoff, and enforce customizable privacy protection.**

***Index Terms*— Social trust, federated learning, differential privacy, federation game.**

Manuscript received 29 December 2022; revised 10 May 2023 and 25 December 2023; accepted 29 February 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Joshi. Date of publication 2 October 2024; date of current version 17 October 2024. This work was supported in part by NSFC under Grant 62302387, Grant U23A20276, Grant U22A2029, and Grant U20A20175; in part by the Postdoctoral Innovative Talent Support Program of China under Grant BX20230282; in part by China Postdoctoral Science Foundation under Grant 2023M732820; in part by Shaanxi Province Postdoctoral Science Foundation under Grant 2023BSHT-BZZ07; and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Zhou Su.)*

Yuntao Wang, Zhou Su, Yanghe Pan, and Tom H. Luan are with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: zhousu@ieee.org).

Ruidong Li is with the Institute of Science and Engineering, Kanazawa University, Kanazawa 920-1192, Japan.

Shui Yu is with the School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia.

Digital Object Identifier 10.1109/TNET.2024.3379439

## I. INTRODUCTION

WITH the explosive growth of smart phones, wearables, and Internet of things (IoT) devices, nearly 75% of data is anticipated to be produced, gathered, and processed outside of clouds by 2025, particularly at distributed end-devices at the edge [1]. Due to data privacy and ownership concerns, aggregating such vast volumes of distributed data into a central cloud for artificial intelligence (AI) model training can be both illegal and privacy risky [2], [3]. Federated learning (FL) offers a promising privacy-preserving AI paradigm that adheres to the principle of bringing code to data, instead of the opposite direction [4], [5], [6]. In FL, individual devices periodically train AI sub-models (e.g., gradients) using local data and send to the aggregation server (e.g., the cloud), which synthesizes a global AI model for next-round training [7]. As users only share the learned model parameters instead of the original raw data, the privacy concerns can be significantly resolved under FL.

However, in such an open and untrusted FL environment, users' privacy can still be divulged from their trained sub-models (e.g., gradients) by sophisticated adversaries and the untrusted server, via attacks such as membership inference [8] and model inversion [9]. For example, experiments in [10] validate that clients' private training data can be stolen from the publicly shared gradients in vision and language tasks. Existing countermeasures mainly rely on the local differential privacy (LDP) techniques [6], [11], [12] due to the strict theoretic guarantees and low computation overhead, in which individuals independently sanitize their sub-models by adding random LDP perturbations, as shown in Fig. 1(a). In LDP-based approaches, the larger injected LDP noise enforces stronger privacy provisions but also entails more severe performance degradation, which eventually deteriorates individual payoff. Thereby, LDP-based FL approaches usually necessitate a tradeoff between privacy and utility. Current efforts mainly focus on designing optimized FL approaches [13], [14], [15], [16], [17] to strive for such a balance, while ignoring the inner and lasting connections among users, such as social relationships.

With the great success of social networks, social ties have been widely established among mobile users. For instance, 2.93 billion social users monthly interacted via Facebook in the first quarter of 2022, with an average of roughly 200 friends per user [18]. Benefiting from large-scale social networks,

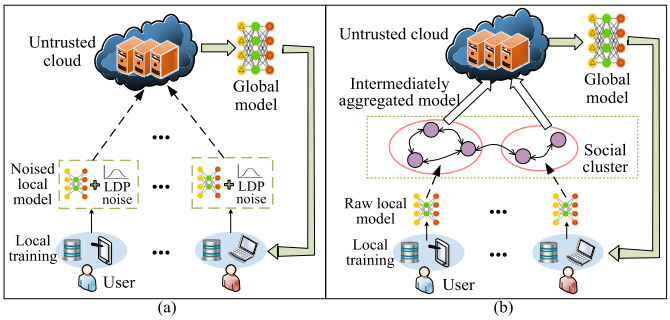



Fig. 1. Illustration of (a) conventional LDP-based FL under an untrusted cloud; (b) the social-aware clustered federated learning (SCFL).

individuals can easily invite their trusted and familiar social friends as cooperative learners and form socially clustered federations (or clusters), as shown in Fig. 1(b). Within each social cluster, members can aggregate their trained sub-models into a combined one before uploading to the cloud. Essentially, as individuals inside each social cluster are mutually trusted, they no longer need to apply LDP perturbations to the trained sub-models, which considerably enhances model utility (e.g., model accuracy) and individual payoff. Meanwhile, both external adversaries and the curious cloud cannot deduce the gradient information of each participating user from the intra-cluster intermediate model aggregations, thus well protecting user privacy. As such, the social-aware clustered federated learning (SCFL) paradigm emerges as a promising strategy to enhance model utility while enforcing privacy protection in FL with low cost and high feasibility.

To practically deploy SCFL services, a series of fundamental challenges still need to be resolved. 1) As users are generally selfish and profit-seeking, they can determine whether or not to join a social cluster, as well as which social cluster to join, depending on the payoffs and social ties. Thereby, *it remains a concern to distributively form a stable and optimized social cluster structure*. 2) Heterogeneous users typically have distinct quality, quantity, and non-IID degree of training data samples in undertaking different FL tasks, resulting in distinct model quality and contributions [7]. Besides, selfish individuals often tend to benefit from the SCFL without contributing to the social cluster, and such free-riding behaviors may lead to poor learning outcomes [19]. *There exists a challenge in contribution quantification and fair revenue allocation within each social cluster with free-rider defense*. 3) In clusters with low mutual social trust, learners may still need to add modest amount of LDP perturbations for privacy concerns. Due to the diversity of users' social ties and privacy preferences, *how to design a flexible and differentiated perturbation mechanism to attain a tradeoff between privacy and utility is a challenging issue*.

To address these issues, this paper proposes a novel and efficient social-aware clustered federated learning (SCFL) scheme with Nash-stable clustering structure, free-rider prevention, and customized privacy preservation, by using a game-theoretical approach. Specifically, we model the interactions among socially connected learners as a distributed federation game with transferable utility (FTU), and formally define the federal payoff and cost of social clusters. Considering heterogenous training samples, data quality, and non-IID degrees of users, we then devise a fair revenue allocation mechanism for all members in each social cluster based on their quantified contributions. Next, for users joining clusters with relatively low trust, a customizable privacy preservation mechanism is designed to meet users' privacy expectations by adaptively determining the privacy protection level depending on both trust-related factors and structural information of the social network. Furthermore, we design an iterative two-sided matching algorithm to derive the Nash-stable social clustering structure, where each individual determines the transfer strategy to affiliate with the optimal cluster while each social cluster determines the optimal admission strategy to accept the optimal learner.

The main contributions of this work are summarized below.

- **Framework**: We propose a novel hierarchical SCFL framework, which realizes low-cost, feasible, and customized FL services by exploiting users' social attributes.
- **Algorithms**: We formulate the optimal social cluster formation problem among individuals with social ties as a FTU game, and devise a suite of algorithms including fair revenue allocation, customizable privacy preservation, and iterative two-sided matching, which converges to Nash-stable equilibrium.
- **Validations**: We implement experiments on real-world datasets to validate the effectiveness of the proposed scheme. Numerical results show that our SCFL can bring higher learning utility and better individual payoff while enforcing customizable privacy protection, compared with existing representatives.

The remainder of the paper is organized as follows. Section II reviews the related works. In Section III, we present the system model. Section IV presents the detailed construction of the proposed SCFL scheme. Performance evaluation is shown in Section V. Section VI concludes this paper and points out the future work.

## II. Related Works

In FL, many works have studied the impact of differential privacy (DP) noises on model performance, assuming that the aggregation server is semi-honest (i.e., honest-but-curious). Besides, many of them strive for a tradeoff between privacy protection and model utility under FL. Zeng et al. [20] study a privacy-enhanced federated temporal difference learning mechanism by injecting DP perturbations to users' shared gradients, where both the privacy bound and the upper bound of utility loss are derived using rigorous analysis. Wei et al. [11] develop an example-level DP algorithm in FL under an untrusted aggregation server, where a dynamically decaying noise-adding strategy is devised for model utility enhancement. Shen et al. [12] design an optimized LDP perturbation method to reduce the impact of LDP noise in FL models via a perturbation regularizer with LDP guarantees for clients. Mohamed et al. [14] propose an optimized user sampling mechanism in DP-based wireless FL environments to balance the size of LDP noise and convergence rate.

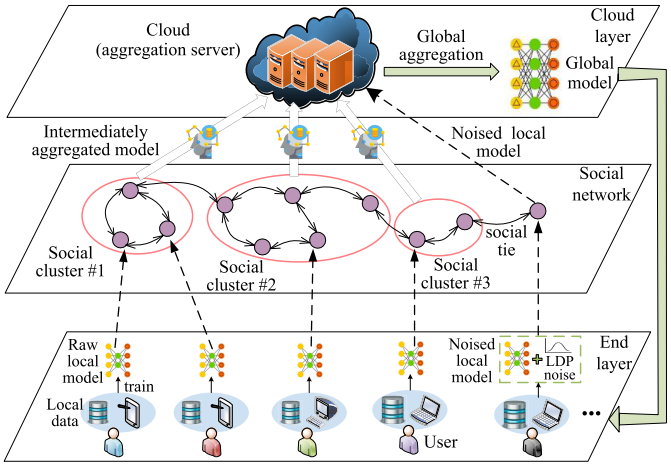



Fig. 2. Illustration of the social-aware clustered federated learning (SCFL).

Wei et al. [16] investigate a multi-agent learning approach to minimize training time and communication rounds in wireless FL settings while enforcing DP for users.

Several works have recently been reported incorporating social effects into collaborative wireless networks such as crowdsensing and FL. Shi et al. [21] exploit social influences among individuals for efficient incentive design in crowdsensing applications, with the aim to maximize the data quality of the crowdsensing platform and cut down the cost of user recruitment for data collection under information asymmetry. By recruiting trustworthy social friends as collaborative learners, Lin et al. [22] design a social-driven incentive mechanism under federated edge learning to minimize the payment of FL service requesters while encouraging edge devices' resource contributions in FL. However, existing works mainly leverage the social influences for reliable participant recruitment and cost-effective incentive design, whereas the use of social attributes and formation of social clusters among users for better privacy-utility tradeoff in DP-based FL are not taken into account.

Distinguished from previous works, our work integrates social effects into FL to simultaneously achieve high model performance and rigorous privacy protection with high feasibility and low cost, via optimized social clustering and customized LDP perturbations.

## III. System Model

In this section, we first introduce the network model and then discuss the design goals of SCFL. Table I summarizes the notations used in the remaining of this paper.

### A. Network Model

As depicted in Fig. 2, our SCFL consists of three layers: the cloud layer, the social layer, and the end layer.

*Cloud layer*. The public cloud (e.g., Azure, AWS, and Google cloud) serves as the global aggregation server in FL, which is assumed to be *semi-trusted* (i.e., *honest-but-curious*) [11], [12]. Namely, the cloud will honestly perform the global model aggregation in each communication round $k$ but is curious about the privacy in user's local model updates. The cloud platform hosts various FL tasks (e.g., image and sentiment classification) to be accomplished.

TABLE I
Summary of Notations

| Notation | Description |
|---|---|
| $\mathcal{N}$ | Set of users with social connections in a FL task. |
| $\Phi$ | Set of disjoint social clusters for users in $\mathcal{N}$. |
| $\phi_j$ | Cluster head of $\Phi_j$. |
| $\Upsilon_{n,j}$ | Social influence of user $n$ in social cluster $\Phi_j$. |
| $\mathcal{G}$ | Social graph of users in $\mathcal{N}$. |
| $\Theta^k$ | Global model at global round $k$. |
| $\Theta_n^k$ | Local model of user $n$ at global round $k$. |
| $\alpha_{n,j}$ | Social trust degree between user $n$ and $\phi_j$. |
| $\alpha_{th}$ | Predefined trust threshold. |
| $e_{n,m}$ | Direct trust between users $n$ and $m$. |
| $\tau_{n,m}$ | Indirect trust between users $n$ and $m$. |
| $K_{n,m}^P$ | Numbers of positive interactions between users $n$ and $m$. |
| $K_{n,m}^N$ | Numbers of negative interactions between users $n$ and $m$. |
| $\nu$ | Penalty factor for negative interactions. |
| $\Gamma_b$ | Exponential time decay effect. |
| $\xi$ | Time decay rate. |
| $T_{path}$ | Shortest social path connecting users $n$ and $m$. |
| $\sigma_{n,j}$ | Scale of DP noise of user $n$ in social cluster $\Phi_j$. |
| $\sigma_{\max}$ | Maximum affordable noise scale. |
| $\epsilon_{n,j}$ | Privacy budget of user $n$ in social cluster $\Phi_j$. |
| $\delta$ | Small failure possibility in DP. |
| $\gamma$ | Concentration factor of Dirichlet distribution. |
| $q_n$ | Quality of local model of user $n$. |
| $\mathscr{L}_n$ | Loss value of user $n$'s trained model. |
| $\mathcal{R}(\Phi_j)$ | Federal revenue of social cluster $\Phi_j$. |
| $\mathcal{C}(\Phi_j)$ | Federal cost of social cluster $\Phi_j$. |
| $\mathcal{V}(\Phi_j)$ | Federal utility of social cluster $\Phi_j$. |
| $\psi_n(\Phi_j)$ | Individual payoff of user $n$ in social cluster $\Phi_j$. |
| $\varsigma$ | Additional reward assigned to each cluster head. |
| $\varpi_{n,j}$ | Weight of member $n$ in social cluster $\Phi_j$. |
| $\mathcal{A}$ | Action space of user in social cluster formation. |
| $\rho_n(.)$ | Preference function for user $n$. |
| $\mathcal{H}_n$ | Historical clusters that have rejected user $n$'s transfer request. |
| $\mathcal{C}_{n,t}^{\text{cluster}}$ | Transferable cluster set of user $n$ at iteration $t$. |
| $\mathcal{C}_{\mathcal{S},t}^{\text{user}}$ | Set of candidate users that tend to join social cluster $\mathcal{S}$. |

*End layer*. The end layer is composed of a set of users interested in participating in FL services, denoted by $\mathcal{N} = \{1, \cdots, n, \cdots, N\}$. For each FL task, each user $n \in \mathcal{N}$ uses the owned smart device (e.g., smart phones, wearables, and smart vehicles) to jointly train a globally shared AI model using their local private data under the FL paradigm, coordinated by the cloud.

*Social layer*. Generally, users are featured with social attributes (e.g., friends, relatives, and classmates) and are interconnected via the social network. In SCFL, users involved in a common task can dynamically form socially clustered disjoint federations, and the set of which is denoted as $\Phi = \{\Phi_1, \cdots, \Phi_j, \cdots, \Phi_J\}$. Within each social cluster $\Phi_j \in \Phi$, mutually trusted peers can directly send their raw local model updates instead of the noised version to the *cluster head*,[1] denoted by $\phi_j$, who then produces an intermediately aggregated model. Here, the member with the highest centrality degree (i.e., social influence $\Upsilon_{n,j}$) in the cluster is selected as the cluster head (details are shown in Sect. IV-B), which

[1]A trusted processor [23] such as Intel SGX and ARM TrustedZone can also act as the cluster head for intra-cluster model aggregation.

is assumed to be *socially trusted*[2] inside the social cluster. After aggregating the local models, the cluster head forwards the combined model to the cloud. As such, both the global model's utility (e.g., model performance) and users' payoffs can be improved. Besides, since neither the curious cloud nor external adversaries can infer any member's raw model update from the intermediate model aggregation, the privacy of learners in each social cluster can be well-protected (details are shown in Sect. IV-D).

In a typical cross-device FL setting, clients only communicate with the central server. Compared with conventional cross-device FL scenarios, our SCFL framework incorporates the social relationships between users and allows socially trusted users to form stable social clusters via social platforms (e.g., Facebook) to improve model utility, which can be applied to general cross-device FL applications.

*Illustrating example*. We take Facebook as an example of the social platform. A FL task publisher announces his/her FL task along with the Facebook group ID, learning model structure, IP address of the cloud aggregation server and etc, on a machine learning community (e.g., Kaggle). Participants can invite their Facebook friends to join the FL process. All participants form a grand Facebook group, within which they can communicate. Then, mutually trusted clients can create multiple disjoint Facebook subgroups. They can freely move between these subgroups or choose to act independently based on individual benefits, until forming a stable partition for all users. Within each subgroup, mutually trusted peers directly share their raw local model updates instead of the noised version with the cluster head, who serves as the manager of the corresponding Facebook subgroup. Each cluster head then forwards the intra-subgroup combined model to the aggregation server who produces a global model via inter-subgroup aggregation. Finally, the global model is distributed back to the grand Facebook group for next-round training.

### B. Design Goals

The target of our SCFL is to simultaneously attain the following design goals.

**1) Dynamically optimized social cluster structure**. Considering users' diverse social ties and dynamic competition and cooperation, SCFL aims to form an optimized social cluster structure to maximize the payoffs of profit-driven users.

**2) Fair revenue allocation with free-rider resistance**. Considering diverse user characteristics (e.g., training samples, data quality, and data distribution) and potential free-riding behaviors in FL [19], SCFL should resist free-riders and ensure a fair division of cluster revenue inside each social cluster.

**3) Customized privacy preservation**. For the social clusters with low mutual trust, SCFL needs to enhance the privacy protection of participants by adding customized LDP noises to their local model updates.

[2]In a social cluster, its security level (e.g., risk of user's privacy leakage) in social-layer model aggregation is associated with the mutual social trust between the user and the cluster head. As the social trust degree may vary over time and across users, the security level of the social cluster can be temporally dynamic and heterogenous for different users.

## IV. SCFL: Social-Aware Clustered Federated Learning Scheme

In this section, we first present the design overview (Sect. IV-A), and then the social trust model (Sect. IV-B) and the federation game (Sects. IV-C), followed by the game analysis (Sect. IV-D) and algorithm design (Sect. IV-E).

### A. Design Overview

The overall objective of SCFL is to learn a global model for all participating clients. The workflow of SCFL consists of five successive phases: (i) social trust evaluation, (ii) social cluster formation, (iii) user-side local model training by all clients, (iv) intra-cluster intermediate model aggregation by the cluster head, and (v) inter-cluster global model aggregation by the cloud.

Phase 1: Social trust evaluation. Each user $n \in \mathcal{N}$ evaluates the social trust degrees of existing/potential social friends in the social graph $\mathcal{G}$ via (7). Details are shown in Sect. IV-B.

Phase 2: Social cluster formation. A group of socially connected users (i.e., $\mathcal{N}$) self-organize into disjoint social clusters (i.e., $\Phi$) depending on the payoffs and social trusts. Particularly, each user $n$ determines which social cluster to join or work alone. For some clusters with low mutual social trust, the participants will add modest LDP noises for privacy concerns. Besides, the members of each cluster independently decide whether to accept the newcomers. Details are shown in Sects. IV-C$\sim$ IV-E.

Phase 3: Local model training. Each user $n$ trains the received global AI model $\Theta^{k-1}$ in previous round $k-1$ using local private data $\mathcal{D}_n$ via stochastic gradient descent (SGD) and produces the local AI sub-model $\Theta_n^k$ in current round:

$$\Theta_n^k \leftarrow \Theta^{k-1} - \eta \nabla \mathscr{L}_n \left(\Theta^{k-1}\right), \tag{1}$$

where $\eta$ is the learning rate and $\mathscr{L}_n$ is the loss function on user $n$'s local data samples. If user $n$ joins a cluster $\Phi_j$ with high social trust (i.e., $\alpha_{n,j} \geq \alpha_{th}$), the raw sub-model $\Theta_n^k$ is directly sent to the cluster head $\phi_j$. If user $n$ joins a cluster with relatively low trust (i.e., $0 < \alpha_{n,j} < \alpha_{th}$), a modestly noised sub-model $\widetilde{\Theta}_n^k$ is sent to $\phi_j$. Here, $\alpha_{n,j}$ is the trust degree between user $n$ and $\phi_j$, and $\alpha_{th} \in (0,1)$ is a predefined public trust threshold, whose value depends on specific FL tasks. Otherwise, user $n$ works alone and uploads the noised model $\widehat{\Theta}_n^k$ injected with the uniform and relatively large LDP noise to the cloud (details are shown in Sect. IV-D1).

Phase 4: Intra-cluster model aggregation. Each social cluster head $\phi_j$ aggregates the local model updates of all the members[3] in the social cluster and uploads the intermediate aggregation outcome to the cloud, i.e.,

$$\begin{cases} \overline{\Theta}_j^k = \sum\nolimits_{n \in \mathcal{N}_j^1} q_{n,j} \Theta_n^k + \sum\nolimits_{n \in \mathcal{N}_j^2} q_{n,j} \widetilde{\Theta}_n^k, & (2) \\ Q_j = \sum\nolimits_{n \in \mathcal{N}_j^1 \bigcup \mathcal{N}_j^2} q_{n,j}. & (3) \end{cases}$$

[3]To resist Byzantine attacks (e.g., model poisoning) of participants, existing Byzantine-resilient aggregation mechanisms [23], [25], [26], [27] in different FL settings can be further applied, which is out of scope of this paper.

In (2) and (3), $\mathcal{N}_j^1$ and $\mathcal{N}_j^2$ are the sets of users that send raw sub-models and noised sub-models to cluster $\Phi_j$, respectively. $q_{n,j}$ is the quality of user $n$'s local model, which is evaluated in Sect. IV-D2. When detailed curve-fitting parameters are unavailable, we adopt an iterative approach in [24] (Sect. IV-A-1) to estimate the quality of client's local model.

Phase 5: Inter-cluster global model aggregation. The cloud synthesizes the intermediate aggregations from various social clusters into the current global model $\Theta^k$ weighted by model utilities, i.e.,

$$\Theta^k \leftarrow \frac{1}{\sum_{\Phi_j \in \Phi} Q_j} \sum\nolimits_{\Phi_j \in \Phi} \overline{\Theta}_j^k. \tag{4}$$

Until the global round $k$ attains its maximum value or the global model obtains the predefined accuracy, the above learning process in phases 3-5 finishes.

### B. Social Trust Evaluation

Let $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}, \mathcal{T} \rangle$ denote the social graph among users in the set $\mathcal{N}$. Here, $\mathcal{E} = \{e_{n,m} | \forall n, m \in \mathcal{N}, n \neq m\}$ is the set of edges between users, and $e_{n,m} \in [0,1]$ denotes the social relationship or social closeness between two users $n$ and $m$ ($n \neq m$). Particularly, $e_{n,m} = 1$ means that two users have the strongest social tie, and $e_{n,m} = 0$ implies that they are strangers. Let $\alpha_{n,m} \in [0,1]$ denote the social trust degree between two users $n$ and $m$, and the set of which is denoted as $\mathcal{T} = \{\alpha_{n,m} | \forall n, m \in \mathcal{N}, n \neq m\}$. The social trust degree $\alpha_{n,m}$ is evaluated based on the direct social closeness and indirect topological relationships [28], [29].

The direct trust $e_{n,m}$ is oriented from the direct experience in historical interactions (e.g., sharing microblogs, photos, videos, and engaging in social gaming),[4] which is affected by the interaction experience and interaction occurrence time. According to [30], we have

$$e_{n,m} = \max\left\{ \frac{\sum_{b=1}^{K_{n,m}^P} \Gamma_b - \nu \sum_{b=1}^{K_{n,m}^N} \Gamma_b}{K_{n,m}^P + K_{n,m}^N}, 0 \right\}, \tag{5}$$

where $K_{n,m}^P$ and $K_{n,m}^N$ are the total numbers of positive and negative interactions[5] between user $n$ and user $m$, respectively. $\nu > 0$ is a penalty factor. $\Gamma_b = \exp\left(-\xi\left(t - t_b\right)\right)$ describes the exponential time decay effect as latest interaction can be more important than older ones, where $t_b$ is the occurrence time of $b$-th interaction and $\xi > 0$ is the decay rate.

As direct interactions between users are often inadequate, combining indirect topological relationships (i.e., friend-of-friend relationships) in the social graph is necessary for comprehensive trust evaluation. Let $T_{path}$ denote the shortest path connecting user $n$ and user $m$ in $\mathcal{G}$, which excludes the direct link. $|T_{path}|$ is called the social distance.[6] The indirect trust can be computed as the aggregated recommendations from his/her friends in $T_{path}$ [31], i.e.,

$$\tau_{n,m} = \prod_{l,k \in T_{path}, l \mapsto k} e_{l,k}. \tag{6}$$

$l \mapsto k$ means that users $l$ and $k$ are adjacent in the path $T_{path}$. Notably, the multiplication of trust values is adopted instead of computing the average (as done in [28]) to reflect the impact of a low trust value on the global aggregation outcome.

Thereby, the global social trust degree can be attained as:

$$\alpha_{n,m} = \omega e_{n,m} + (1 - \omega)\tau_{n,m}, \tag{7}$$

where $\omega \in [0,1]$ is the weight factor. Notably, $\alpha_{n,m} \in [0,1]$. Besides, as the social connections between users are temporally evolutionary, the social trust $\alpha_{n,m}$ between users is dynamically evaluated. Denote $\Upsilon_{n,j}$ as the centrality degree (or social influence) of user $n$ in social cluster $\Phi_j$, i.e., the number of neighbors that user $n$ has in cluster $\Phi_j$. Here, $\Upsilon_{n,j} = \sum_{l \in \Phi_j} f_{n,l}$, where $f_{n,l} = \{0,1\}$ and $f_{n,l} = 1$ if $e_{n,l} > 0$. Otherwise, $f_{n,l} = 0$.

### C. Federation Game Formulation

In federation game, the mutual communication capability among all players is a basic assumption, which the social network in our work can enable. The social cluster formation process among social individuals is formulated as a federation game with transferable utility (FTU), where socially connected learners can self-organize into disjoint social clusters for maximized individual payoffs.

*Definition 1 (FTU Game):* For every FL task, a FTU game is formally defined by a 4-tuple $(\mathcal{N}, \Phi, \mathcal{V}, \mathcal{A})$, which includes the following key components.

- *Players:* The game players are a set of social users involved in the FL task (i.e., $\mathcal{N}$).
- *Federation structure:* A partition structure, denoted as $\Phi = \{\Phi_1, \cdots, \Phi_J\}$, divides the player set $\mathcal{N}$ into mutually disjoint clusters such that $\Phi_j \cap \Phi_{j'} = \emptyset$, $\forall j \neq j', \forall i \in \mathcal{I}$, and $\cup_{j=1}^J \Phi_j = \mathcal{N}$.
- *Payoff:* The federal payoff of each social cluster $\mathcal{S} \subseteq \mathcal{N}$, denoted as $\mathcal{V}(\mathcal{S})$, can be arbitrarily apportioned among all the members within $\mathcal{S} \in \Phi$. The individual payoff of each player $n \in \mathcal{N}$ that joins in a cluster $\mathcal{S} \in \Phi$ is denoted as $\psi_n(\mathcal{S})$.
- *Strategy:* The action space of each player is denoted as $\mathcal{A}$. Each player can determine either to act alone by applying the *solo training* strategy or join a social cluster to jointly produce a intra-cluster aggregated model using the *clustered training* strategy.

Next, we define group rationality and individual rationality. Based on them, the core of the FTU game is defined.

*Definition 2:* A payoff vector $\boldsymbol{\psi} = \{\psi_n\}_{n=1}^N$ is said to be *group rational* if $\sum_{n=1}^N \psi_n = \mathcal{V}(\mathcal{N})$. Besides, $\boldsymbol{\psi}$ is said to be *individual rational* if $\psi_n \geq \mathcal{V}(\{n\}), \forall n \in \mathcal{N}$, i.e., the payoff of any user in the FTU game is no less than what they would receive from acting alone.

*Definition 3:* The *core* of the FTU game is a set of stable payoff vectors satisfying both group rationality and individual

[4] It is assumed that for all users engaging in a common FL task, their social trust degrees do not update until a Nash-stable partition in Alg. 1 is formed.

[5] The positive and negative interactions are determined based on user's subjective feelings (e.g., giving a subjective rating to his/her peer) during each online or offline interaction, which may not be symmetric.

[6] We set $|T_{path}| = 2$ to obtain the social recommendation only from his/her friends to preserve user privacy to a large extent. If multiple paths share the same social distance, the indirect trust score is computed by averaging the aggregated recommendations on these pathes. Notably, the computing of indirect trust between two users involves all their common friends in $\mathcal{G}$, regardless of their participation status in the FL task.

rationality, i.e.,

$$\mathcal{C}=\left\{\boldsymbol{\psi}|\sum_{n=1}^{N}\psi_n=\mathcal{V}(\mathcal{N})\,\&\sum_{n\in\mathcal{S}}\psi_n\geq\mathcal{V}(\mathcal{S}),\quad\forall\mathcal{S}\subseteq\mathcal{N}\right\}. \tag{8}$$

In (8), $\sum_{n\in\mathcal{S}}\psi_n\geq\mathcal{V}(\mathcal{S})$ means that players have no incentives to form another cluster $\mathcal{S}$ and reject the proposed $\boldsymbol{\psi}$. A non-empty core implies that participants are incentivized to form the grand federation (i.e., $\{N\}$).

### D. Federal and Individual Payoff Analysis

*1) Customized Local Perturbation:* In the case that a user joins a specific cluster with relatively low mutual social trust, the user may still need to add a modest amount of LDP perturbations for privacy concerns. In most previous works [4], [11], [20], it is supposed that all users are subject to a uniform level of privacy protection, which rules out users' personalized privacy preferences. Here, we develop a trust-oriented customized local perturbation mechanism to satisfy individual privacy expectations in practical scenarios. Particularly, the Gaussian mechanism is adopted by adding artificial noise following the Gaussian distribution $\mathbb{G}(0,\sigma^2S^2)$. The variance $\sigma$ controls the scale of noise. Based on the moments accountant method [32], to preserve $(\epsilon,\delta)$-LDP, the noise scale should satisfy [12]:

$$\sigma\geq\frac{\sqrt{2\log(1.25/\delta)}}{\epsilon}. \tag{9}$$

In (9), $\epsilon>0$ is the privacy budget, and a smaller $\epsilon$ enforces stronger privacy protection. $\delta$ is a small failure possibility (we set $\delta=10^{-6}$). Besides, $S=\max_{\mathcal{D},\mathcal{D}'}||f(\mathcal{D})-f(\mathcal{D}')||_2$ is the L2-sensitivity of query function $f$ on two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Let $\lambda_s$ be the sampling rate of user's local data samples. In the following theorem, we show the privacy amplification property of DP.

*Theorem 1 [33]:* According to the privacy amplification property, the Gaussian mechanism with sub-sampling ensures $(\epsilon',\lambda_s\delta)$-LDP and guarantees stronger privacy preservation, where

$$\epsilon'=\log(1+\lambda_s(\exp(\epsilon)-1)). \tag{10}$$

*Remark:* Theorem 1 shows that it provisions stronger privacy preservation by applying the DP mechanism on a random subset of participant's local data samples than on the entire dataset. Moreover, Theorem 1 indicates that the added differentiated Gaussian noises strictly enforce LDP and can preserve participants' privacy in social-layer model aggregation process.

Whenever a user $n$ tends to join a cluster $\Phi_j$ with $\alpha_{n,j}\in(0,\alpha_{th})$, our mechanism returns a sanitized AI sub-model $\widetilde{\Theta}_n^k$ in which the chosen privacy level depends on his/her social trust degree $\alpha_{n,j}$ with the corresponding cluster head $\phi_j$. Specifically, the customizable privacy budget level in LDP can be linearly mapped based on corresponding trust degree, i.e.,

$$\epsilon_{n,j}=\theta_1\cdot\frac{\alpha_{n,j}}{\alpha_{n,j}+\theta_2}, \tag{11}$$

where $\theta_1$ and $\theta_2$ are positive adjustable coefficients.

*2) Local Update Quality Evaluation:* In SCFL, users usually have distinct data sizes and distributions, as well as the injected Gaussian noise scales on local model updates when joining clusters with relatively low trust. Typically, the lower scale $\sigma_{n,j}$ of injected Gaussian noise, the better performance of the trained model. Besides, as validated in [34], non-IID data can cause performance deterioration in FL compared with IID data. We consider the *label- and quantity-skewed non-IID setting* [35] and focus on *classification tasks* under the FL paradigm. In the literature, the Dirichlet distribution has been widely adopted for dataset partition with both quantity and label distribution shifts under the non-IID environment for FL classification tasks, such as [36] for image classification tasks and [35] for text classification tasks. Hence, we employ the Dirichlet distribution to characterize the heterogeneity of data size and data distribution among FL participants. Consider a classification task with $Y$ classes, where training examples of each client is drawn from a Dirichlet distribution parameterized by a vector $\boldsymbol{a}\sim\mathrm{Dir}(\boldsymbol{\gamma})$ with the following probability density function (PDF):

$$p(\boldsymbol{a}|\boldsymbol{\gamma})=\frac{1}{\mathcal{B}(\boldsymbol{\gamma})}\prod_{y=1}^{Y}a_y^{\gamma_y-1},\ \gamma_y>0,\ \sum_{y=1}^{Y}a_y=1. \tag{12}$$

The multivariate beta function $\mathcal{B}(\boldsymbol{\gamma})\triangleq\frac{\prod_{y=1}^{Y}\Gamma(\gamma_y)}{\Gamma(\sum_{y=1}^{Y}\gamma_y)}$ is the normalization constant. $\gamma_y>0,\forall y\in[1,Y]$ is a concentration factor controlling the identicalness among participants. If $\gamma_y\to\infty,\forall y$, all users have identical distributions. If $\gamma_y\to 0,\forall y$, each user only holds one class of samples at random. For simplicity, we set $\gamma_y=\gamma,\forall y$.[7]

Real-world experimental results on the MNIST dataset in Sect. V-B show that the test loss value $\mathscr{L}_n$ of the distributively trained model[8] at the end of training (i.e., when the communication rounds reach the maximum value) can be well-suited by the 3D sigmoid curve with respect to user $n$'s noise scale $\sigma_{n,j}$ and non-IID degree $\gamma$:

$$\mathscr{L}_n=\mathscr{L}(\sigma_{n,j},\gamma)=\frac{\mu_1\exp(-\mu_2\cdot\gamma)}{\mu_3+\exp(-\mu_4\cdot\sigma_{n,j})}+\mu_5, \tag{13}$$

where $\mu_1,\cdots,\mu_5$ are positive curve fitting parameters.[9] The loss in Eq (13) is an empirical fit to the actual loss. The

[7]In practice, the value of $\gamma$ can be estimated by the aggregation server before performing the FL task. For example, a questionnaire can be sent to all participating users, which collects their data distribution (including the label classes of local data and the corresponding data amount for each class). After computing the frequency of the class for each user's dataset, the approximate value of $\gamma$ in Dirichlet distribution can be estimated via parameter estimation methods such as maximum likelihood estimation [39], [40].

[8]When clients adopt different DP noise scales under FL, it becomes very challenging to directly obtain the theoretical relationship between the local model quality of client $n$ and its noise scale. As an alternative, we use the quality of the global model, where all clients adopt the same DP noise scale $\sigma_{n,j}$, to represent the local model quality of client $n$ with noise scale $\sigma_{n,j}$.

[9]As the curve-fitting parameters $\mu_i$ are fixed and uniform for all users involved in a common FL task, the stable social cluster structure produced by the federated game algorithm in Alg. 1 does not depend on the detailed values of $\mu_i$ but the form of curve-fitting function. As the specific form of curve-fitting function $\mathscr{L}_n$ can be a priori for a given FL task, we can leverage the historical knowledge for various FL tasks in the public FL market [37]. Specifically, 1) if the target FL model exists in the historical FL tasks, we directly apply the corresponding form of curve-fitting function; 2) otherwise, we can select a historical FL task that closely resembles the target task, and employ the corresponding curve-fitting form.

numerator part $\mu_1 \exp(-\mu_2 \cdot \gamma)$ reflects the diminishing marginal loss value when the concentration factor $\gamma$ increases. The denominator part $\mu_3 + \exp(-\mu_4 \cdot \sigma_{n,j})$ captures that the increasing noise scale $\sigma_{n,j}$ results in a performance degradation. Notably, curve fitting is a typical approach to determine the AI model quality, and a similar manner has been applied in [37], [38]. In the experiments in Sect. V-B, the loss function in (13) fits well when $\sigma_{n,j}$ falls in $[0, \sigma_{\max}]$, where $\sigma_{\max}$ is the maximum tolerable noise scale to guarantee model availability in practical FL services. It is because that oversized noise can completely distort model parameters and reduce model accuracy to the level of random inference.

Typically, the lower the loss value, the better the model performance. Moreover, the faster the drop rate of the loss, the faster the ascent rate of the model utility (e.g., model accuracy). Hence, the model utility function $q(\mathscr{L}_n)$ should meet $q(\mathscr{L}_n) > 0$ and $\frac{\mathrm{d}q(\mathscr{L}_n)}{\mathrm{d}\mathscr{L}_n} < 0$. Via curve fitting approaches, the quality function with respect to the loss value is formulated in the linear form to meet the above requirements, i.e.,

$$q_n = q(\mathscr{L}_n) = -\kappa_1 \cdot \mathscr{L}_n + \kappa_2, \tag{14}$$

where $\kappa_1 > 0$ is a positive adjustment factor. The factor $\kappa_2$ captures the maximum model utility when the loss $\mathscr{L}_n \to 0$.

*3) Federal Payoff Function:* The total revenue of social cluster $\Phi_j$ is related to the overall utility of immediate model aggregations. For simplicity, the federal revenue is computed based on the sum of model utilities of all its members [41]:

$$\mathcal{R}(\Phi_j) = \lambda_p \sum\nolimits_{n \in \Phi_j} q_n, \tag{15}$$

where $\lambda_p$ is the task publisher's unit payment per model quality. $q_n$ is the quality of local model (QoLM) of user $n$ in cluster $\Phi_j$ based on (13) and (14). Besides, individuals can apply the solo training strategy by forming a singleton, namely, $\Phi_j = \{n\}$. In this case, the federal revenue is computed as $\mathcal{R}(\{n\}) = \lambda_p \widehat{q}_n$, where $\widehat{q}_n$ is computed via $\sigma_{n,j} = \sigma_{\max}$.

Users within each social cluster need to frequently communicate with the cluster head for intra-cluster model aggregation. Based on [42] and [43], the federal cost $\mathcal{C}(\Phi_j)$ can be measured by the communication overhead that is proportional to the cluster size $|\Phi_j|$, i.e.,

$$\mathcal{C}(\Phi_j) = \begin{cases} \lambda_c |\Phi_j|, & \text{if } |\Phi_j| > 1, \\ 0, & \text{if } |\Phi_j| = 1, \end{cases} \tag{16}$$

where $\lambda_c$ is a positive scaling coefficient.

The federal utility of social cluster $\Phi_j$ can be denoted as the total revenue minuses the cost:

$$\begin{aligned} \mathcal{V}(\Phi_j) &= \mathcal{R}(\Phi_j) - \mathcal{C}(\Phi_j) \\ &= \begin{cases} \lambda_p \sum\limits_{n \in \Phi_j} q_n - \lambda_c |\Phi_j|, & \text{if } |\Phi_j| > 1, \\ \lambda_p \widehat{q}_n, & \text{if } |\Phi_j| = 1. \end{cases} \end{aligned} \tag{17}$$

*4) Fair Payoff Division Within Cluster:* The proportional fairness is employed for fair payoff division within each social cluster while conserving individual rationality, in which the *extra payoff* is divided into weights based on participants' non-cooperative payoffs. The individual payoff of user $n \in \Phi_j$ is given by

$$\psi_n(\Phi_j) = \varpi_{n,j} \left( \mathcal{V}(\Phi_j) - \sum\nolimits_{l \in \Phi_j} \mathcal{V}(\{l\}) - \varsigma \right) + \mathcal{V}(\{n\}), \tag{18}$$

where $\mathcal{V}(\{l\})$ and $\mathcal{V}(\{n\})$ denote the non-cooperative payoffs of users $l$ and $n$, respectively. $\varpi_{n,j} = \frac{q_n}{\sum_{l \in \Phi_j} q_l}$ is the weight of member $n$ in cluster $\Phi_j$. In (18), users with higher QoLMs deserve more extra revenue. Besides, as the cluster head is responsible for intra-cluster model aggregation and result uploading, it consumes more computation, storage, and communication resources than other individuals within the cluster. Thereby, an additional reward $\varsigma$ is assigned to the cluster head as an incentive.

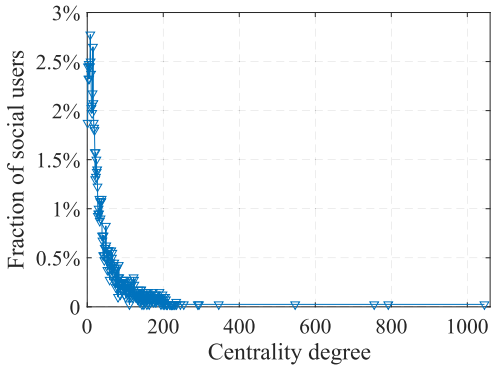



Fig. 3. Node centrality degree distribution of Facebook network [44].

*Remark:* When performing different FL tasks, as participants usually have different data sizes, data quality, and data distributions, they usually form different social clusters.

Next, we show that the grand social cluster $\{N\}$ which contains all users in $\mathcal{N}$ will not form for a FL task. In the FTU game, the federal cost of a cluster grows as the cluster size increases, which can be considerably high for the grand federation. Besides, it is usually impractical for all users to maintain a close social connection with the cluster head $\phi$ of the grand federation. As depicted in Fig. 3, experiments on the real-world Facebook network [44] with 4039 nodes and over 88K edges show that the top-4 users with the highest centrality have 1045, 792, 755, and 547 social neighbors, respectively, and similar results also apply for other social networks such as Google+ and Twitter networks. It implies that social users generally belong to different social communities. If the grand federation exists, due to the relatively low social trust with the cluster head $\phi$, most users still need to add modest LDP noise via (11) in the grand federation, which eventually deteriorates model utility and individual payoff. As such, part of users tend to deviate from the grand coalition and form several disjoint clusters. Hence, the grand federation is unstable.

*Remark:* According to Definition 3, as the grand social cluster will not form, the core of $(\mathcal{N}, \Phi, \mathcal{V}, \mathcal{A})$-FTU game is empty. It indicates that social users have no incentives to form a grand federation. In the following, we devise a distributed cluster formation algorithm to derive such stable disjoint clusters.

**Algorithm 1** Distributed Stable Social Cluster Formation

**Input**: $\mathcal{G}$, $\mathcal{N}$, $\mathcal{V}$, $\mathcal{A}$, $\gamma$, $\sigma_{\max}$, $\alpha_{th}$, $\omega$
**Output**: $\Phi^*$, $\psi = \{\psi_n\}_{n\in\mathcal{N}}$
1 **Initialize:** $t=0$, $\Phi=\Phi^{(0)}$, $\mathcal{C}_{n,t}^{\text{cluster}}=\emptyset$, $\mathcal{C}_{\mathcal{S}}^{\text{user}}=\emptyset$, $\mathcal{H}_n=\emptyset$;
2 **while** $\rho_n(\mathcal{S}') < \rho_n(\mathcal{S}), \forall n \in \mathcal{S} \subseteq \mathcal{N}, \forall \mathcal{S}' \in \Phi^{(t)} \cup \{\emptyset\}$ **do**
3  **for** $n \in \mathcal{N}$ **do**
4  Update $\mathcal{H}_n$;
5  **for** $\Phi_j^{(t)} \in \Phi^{(t)} \setminus \mathcal{H}_n$ **do**
6  Compute $\alpha_{n,j}$ via (7);
7  **if** $0 < \alpha_{n,j} < \alpha_{th}$ **then**
8  Compute $\epsilon_{n,j}$ via (11) and $\sigma_{n,j}$ via (9);
9  **if** $\alpha_{n,j} = 0$ *or* $\Phi_j^{(t)} = \{\emptyset\}$ **then**
10  Set $\sigma_{n,j} = \sigma_{\max}$;
11  **else**
12  Set $\sigma_{n,j} = 0$;
13  Compute $q_n$ via (13) and (14);
14  Compute $\rho_n(\mathcal{S})$ via (20) and $\mathcal{C}_{n,t}^{\text{cluster}}$ via (21);
15  Send transfer request to the cluster $\mathcal{S}^* \in \mathcal{C}_{n,t}^{\text{cluster}}$ via the transfer rule and membership leaving rule;
16  **for** $\mathcal{S} \in \Phi^{(t)}$ **do**
17  Store the users that request to transfer to it in $\mathcal{C}_{\mathcal{S},t}^{\text{user}}$;
18  Accept the most preferred user $n^* \in \mathcal{C}_{\mathcal{S},t}^{\text{user}}$ via the admission rule and membership joining rule while reject other candidates;
19  Do split-and-merge operation via Definition 9;
20  $t = t + 1$;

### *E. Distributed Stable Social Cluster Formation*

The solution of the FTU game is to find a stable clustering structure $\Phi^*$, where the stable partition structure can be acquired via exhaustive searching [4]. However, with more users involved, the number of possible partition iterations increases exponentially [42]. Alternatively, Algorithm 1 shows an iterative two-sided matching algorithm with low complexity to distributively attain the optimal partition strategy, consisting of three steps as below.

Step 1: Partition initialization (line 1). For each FL task, the initial partition $\Phi^{(0)}$ at $t = 0$ depends on specific applications, such as the stable partition results of the previously completed FL mission. When previous partitions are not available, the initial social clustering structure is set as $\Phi^{(0)} = \{1, 2, \cdots, N\}$, where each user forms a singleton [45], [46].

Step 2: Transfer strategy of each user (lines 3–15). Given the current partition $\Phi^{(t)} = \{\Phi_1^{(t)}, \cdots, \Phi_J^{(t)}\}$, every user faces three options: (i) split from the current cluster and work alone by adding LDP noise with scale $\sigma_{\max}$ (i.e., solo training); (ii) split from the current cluster and merge with any other non-empty cluster (if $\alpha_{n,j} \in (0, \alpha_{th})$, a modest amount LDP noise will be added via (11)); (iii) stay in the current cluster. The latter two are clustered training strategies. Besides, to prevent the strategic behaviors of participants and social clusters for fairness and partition stability concerns, the following two membership rules for leaving and joining a social cluster are introduced to restrict users' leaving and joining behaviors within each social cluster.

*Definition 4 (Membership Rule):* The membership rules include the leaving rule and joining rule:

- *Leaving rule*: At iteration $t$, if a social cluster $\mathcal{S} \in \Phi^{(t)}$ decides to admit a new member, then all its current members cannot leave $\mathcal{S}$ at iteration $t$;
- *Joining rule*: At iteration $t$, if any member of a social cluster $\mathcal{S} \in \Phi^{(t)}$ leaves, then this social cluster cannot admit any new user at iteration $t$.

Next, we define each user's preference order and transferable cluster set.

*Definition 5 (Preference Order):* The preference order $\succeq_n$ for any user $n \in \mathcal{N}$ is a transitive and complete relation between two transferable social clusters $\mathcal{S}_1$ and $\mathcal{S}_2$ such that:

$$\mathcal{S}_1 \succeq_n \mathcal{S}_2 \Leftrightarrow \rho_n(\mathcal{S}_1) \geq \rho_n(\mathcal{S}_2). \tag{19}$$

Similarly, for the strict preference order $\succ_n$, we have $\mathcal{S}_1 \succ_n \mathcal{S}_2 \Leftrightarrow \rho_n(\mathcal{S}_1) > \rho_n(\mathcal{S}_2)$. Here, $\rho_n(.)$ is the *preference function* for any user $n \in \mathcal{N}$, $n \notin \mathcal{S}$ and any candidate transferable cluster $\mathcal{S} \in \Phi^{(t)}$, which is defined as:

$$\rho_n(\mathcal{S}) = \begin{cases} \psi_n(\mathcal{S} \cup \{n\}), & \text{if } \psi_l(\mathcal{S} \cup \{n\}) \geq \psi_l(\mathcal{S}), \forall l \in \mathcal{S}, \\ & \& \ \mathcal{S} \notin \mathcal{H}_n \text{ or } \mathcal{S} \neq \emptyset; \\ -\infty, & \text{otherwise.} \end{cases} \tag{20}$$

$\mathcal{H}_n$ denotes user $n$'s history set which stores the historical clusters that he/she has revisited and been rejected. As any user can always revert to form a singleton, $\mathcal{H}_n$ is only applicable to clusters whose size is greater than one.

*Definition 6 (Transferable Cluster Set):* For each user $n \in \mathcal{S}$, its transferable cluster set at iteration $t$ is defined as:

$$\mathcal{C}_{n,t}^{\text{cluster}} = \left\{ \mathcal{S}' | \rho_n(\mathcal{S}') \geq \rho_n(\mathcal{S}), \forall \mathcal{S}' \in \Phi^{(t)} \cup \{\emptyset\} \right\}. \tag{21}$$

*Remark:* If user $n$ had been rejected by a cluster $\mathcal{S}'$ (i.e., $\mathcal{S}' \in \mathcal{H}_n$), the cluster $\mathcal{S}'$ will not occur in user $n$'s transferable cluster set. If $\rho_n(\mathcal{S}') \leq \rho_n(\mathcal{S}), \forall \mathcal{S}' \in \Phi^{(t)} \cup \{\emptyset\}$, there is no transferable cluster nor the empty set to transfer for user $n \in \mathcal{S}$, implying that he/she will stay in the current cluster $\mathcal{S}$. Otherwise, user $n$ decides the transfer strategy according to the following transfer rule.

*Definition 7 (Transfer Rule):* Each user $n \in \mathcal{S}$ sends a merge request to the optimal candidate cluster $\mathcal{S}^* \in \mathcal{C}_{n,t}^{\text{cluster}}$ with the largest preference value, i.e., $\mathcal{S}^* = \arg\max \rho_n(\mathcal{S}'), \forall \mathcal{S}' \in \mathcal{C}_{n,t}^{\text{cluster}}$.

*Remark:* If $\mathcal{S}^* = \{\emptyset\}$, user $n$ prefers splitting from the current cluster $\mathcal{S}$ and forming a singleton. Otherwise, user $n$ prefers merging with another cluster $\mathcal{S}^*$ by splitting from the current cluster $\mathcal{S}$.

Step 3: Admission strategy of each social cluster (lines 16–19). Notably, the transfer order can affect the federal payoffs and the partition result when multiple users ask to join the same cluster $\mathcal{S}$. The following admission rule describes the preference of each cluster for the transfer order.

*Definition 8 (Admission Rule):* For each social cluster $\mathcal{S} \in \Phi^{(t)}$, when it receives multiple transfer requests from multiple candidate users in $\mathcal{C}_{\mathcal{S},t}^{\text{user}}$, it only permits the candidate $n^*$ with the largest preference value, i.e., $n^* = \arg\max \rho_n(\mathcal{S}), \forall n \in \mathcal{C}_{\mathcal{S},t}^{\text{user}}$, and rejects other candidates in $\mathcal{C}_{\mathcal{S},t}^{\text{user}} \setminus \{n^*\}$.

By applying the admission rule, each social cluster $\mathcal{S} \in \Phi^{(t)}$ makes its admission strategy. Then, the following split-and-merge operation is executed for each permitted user, which results in a new partition structure, i.e., $\Phi^{(t)} \rightarrow \Phi^{(t+1)}$.

*Definition 9 ( Split-and-Merge Operation):* A split-and-merge operation that transfers user $n^* \in \mathcal{S}$ to another cluster $\mathcal{S}'$ consists of a split operation (i.e., $\mathcal{S} \triangleright \{\mathcal{S}^-, \{n^*\}\}$) and a subsequent merge operation (i.e., $\{\mathcal{S}', \{n^*\}\} \triangleright \mathcal{S}'^+$). Here, $\mathcal{S}^- = \mathcal{S} \backslash \{n^*\}$ and $\mathcal{S}'^+ = \mathcal{S}' \cup \{n^*\}$.

The above steps 2-3 end until reaching a final Nash-stable partition structure $\Phi^*$ (line 2).

*Definition 10 (Nash-Stability):* A partition $\Phi$ is Nash-stable if $\mathcal{S} \succeq_n \mathcal{S}', \forall n \in \mathcal{S} \subseteq \mathcal{N}, \forall \mathcal{S}' \in \Phi \cup \{\emptyset\}$.

*Theorem 2:* The partition outcome $\Phi^*$ in Alg. 1 is Nash-stable.

*Proof:* We first prove that Alg. 1 can always converge to a final disjoint partition $\Phi^*$, given an arbitrary initial structure $\Phi^{(0)}$. By inspecting the preference function in (20), we can observe that each single split-and-merge operation either results in: (i) an unvisited new partition; or (ii) a singleton with a non-cooperative user. For case (i), as the maximum number of partitions among users in $\mathcal{N}$ is finite and can be obtained by the well-known Bell number function [47], the number of transformations in $\{\Phi^{(0)} \rightarrow \cdots \rightarrow \Phi^{(t)} \rightarrow \cdots \rightarrow \Phi^{(T)} = \Phi^*\}$ is finite. For case (ii), in the next iteration $t+1$, the non-cooperative user should either remain non-cooperative or join a new cluster (which yields an unvisited partition). In all cases, the transformation sequence will terminate after $T$ turns and converge to a final outcome $\Phi^*$.

Next, we prove the Nash-stability by contradiction. Assume that the final partition $\Phi^*$ in Alg. 1 is not Nash-stable. As such, there is a user $n \in \mathcal{S}$ and a cluster $\mathcal{S}' \in \Phi^* \cup \{\emptyset\}, \mathcal{S}' \neq \mathcal{S}$ such that $\mathcal{S}' \succeq_n \mathcal{S}$. Thereby, user $n$ will split from the current cluster $\mathcal{S}$ and merge with $\mathcal{S}'$, contradicting with the fact that $\Phi^*$ is the convergence outcome of Alg. 1. ∎

*Remark:* Theorem 2 indicates that any final partition derived from Alg. 1 is Nash-stable and individually rational. Namely, no user $n \in \mathcal{N}$ tends to leave the current cluster $\Phi_j^*$ and switch to another cluster $\Phi_l^* \in \Phi^* \cup \{\emptyset\}$, $j \neq l$ to improve his/her individual payoff. Notably, the Nash-stable partition outcome $\Phi^*$ produced by Alg. 1 is not unique. For example, different initial partitions may result in distinct partition outcomes. The overall computational complexity of Alg. 1 is $\mathcal{O}(T \cdot N \cdot |\Phi^i|)$ in the worst case. Besides, simulation results from Fig. 16 and Fig. 17 show that our proposed Alg. 1 can quickly converge to the Nash-stable partitions. It indicates that our SCFL only incurs small additional overheads when the social cluster changes (i.e., additions or subtractions).

## V. Experimental Validation

In this section, we conduct extensive experiments using the real Facebook social network and classic MNIST/CIFAR-10 dataset on a workstation with Intel Xeon Platinum 8280 CPU (2.7GHz/4.0GHz), 256G RAM, and two Nvidia GeForce RTX 3090 GPUs. We use PyTorch to implement the SCFL.

### A. Experiment Setup

**Datasets and Models.** We evaluate SCFL on the Facebook ego network [44] with 4039 nodes (i.e., social users) and over 88K edges (i.e., social relations), which shows a real social network topology. The participants are randomly chosen from the Facebook network with varying numbers, i.e., $[50, 100, 150, 200, 250]$, where their social connections are extracted from the Facebook network (i.e., whether there exists an edge between them). As the Facebook ego network does not offer the social strength of these edges, we synthesize the social relationship $e_{n,m}$ of socially connected users via a truncated normal distribution as in [48]. The asymmetric social relationship is possible, i.e., $e_{n,m} \neq e_{m,n}$. Two typical datasets for FL tasks are considered, i.e., the MNIST dataset[10] for handwritten digits recognition and the CIFAR-10 dataset[11] for image recognition. For dataset partition among individuals, the non-IID degree of users' local dataset is controlled by varying the Dirichlet parameter $\gamma$ (as analyzed in Sect. IV-D). The value of $\gamma$ is selected between $[0.05, 20]$. For local model training, the 4-layer CNN model is applied for MNIST, while the 5-layer CNN model is adopted for CIFAR-10. The total numbers of communication rounds are set as 30 and 100 in MNIST and CIFAR-10, respectively. For both MNIST and CIFAR-10, the mini-batch SGD with learning rate $\eta = 0.05$, local batch size 64, and local epoch 1 are adopted for all users. For the same FL task, the hyperparameters including communication rounds, learning rate, batch size, and number of participants are same.

**LDP Noise Adding.** For customized LDP perturbation, we set $\alpha_{th} = 0.7$, $\theta_1 = 100$ (MNIST), $\theta_1 = 200$ (CIFAR10), $\theta_2 = 1$, and $\delta = 10^{-6}$ in (11) to map the social trust $\alpha_{n,j}$ to the privacy protection level $\epsilon_{n,j}$ for privacy-utility tradeoff under the clustered training strategy. We set $\sigma_{n,j} = \frac{\sqrt{2\log(1.25/\delta)}}{\epsilon_{n,j}}$ in (9) based on [12] and [32]. The Gaussian noise scale under the solo training strategy is set as $\sigma_{\max} = 0.6$ for MNIST and $\sigma_{\max} = 0.3$ for CIFAR-10, respectively.

**Federation Game.** For federation game model, we set $\omega = 0.8$, $\varsigma = 30$, $\lambda_p = 0.52$, $\lambda_c = 1.2$, $\kappa_1 = 35.4278$, $\kappa_2 = 102.2444$. To evaluate the effect of social attributes in SCFL, for socially connected users in the real Facebook ego network, we further set the following three levels of social effects. For *strong social effects*, the mutual social trust values between socially connected users are larger than the threshold $\alpha_{th}$; while for *weak social effects*, users' social trust values are randomly distributed within $[0, \alpha_{th}]$. For *no social effects*, there exist no social connections among users, and our strategy automatically degenerates to the typical cross-device FL setting with non-cooperative users.

The performance of SCFL is evaluated by comparing with the following conventional schemes.

- **Uniform DP scheme [22].** In [22], the local model updates of social users are sanitized by adding LDP noise with the uniform scale before global aggregation. As users usually have distinct privacy expectations, the

[10] http://yann.lecun.com/exdb/mnist

[11] https://www.cs.toronto.edu/kriz/cifar.html

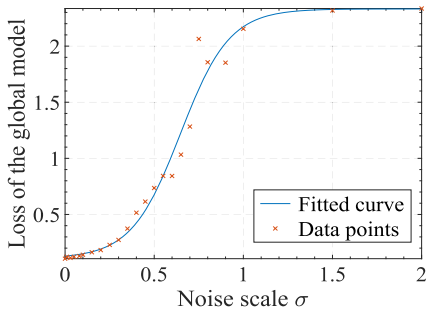



Fig. 4. Curve fitting of loss $\mathscr{L}$ w.r.t noise scale $\sigma$ under IID: $\mathscr{L} = \frac{0.0225}{0.010+\exp(-6.9672\sigma)} + 0.09$.

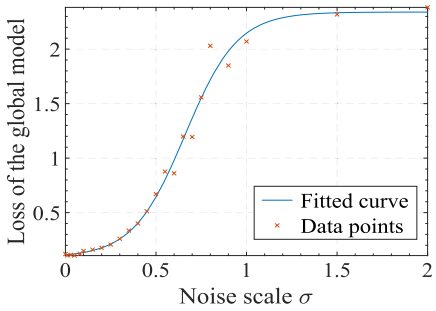



Fig. 5. Curve fitting of loss w.r.t noise scale in non-IID ($\gamma = 0.6$): $\mathscr{L} = \frac{0.02}{0.009+\exp(-7.278\sigma)} + 0.109$.

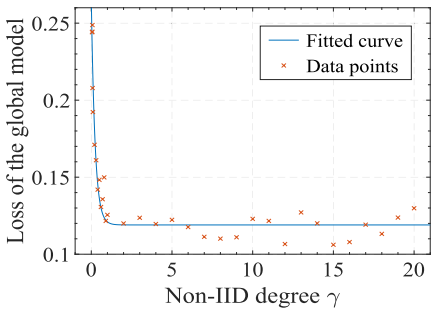



Fig. 6. Curve fitting of loss w.r.t non-IID degree $\gamma$ with $\sigma = 0.1$: $\mathscr{L} = 0.147\exp(-4.288\gamma) + 0.119$.

relatively large LDP noise (i.e., $\sigma_{\max}$) is applied in practice to meet the requirements of most users. Besides, we set $\gamma = 0.6$ for MNIST and $\gamma = 1$ for CIFAR-10.

- **Non-cooperative scheme**. In most works on LDP-based FL such as [12] and [15], individuals apply the solo training strategy and act as singletons in conducting FL tasks.
- **Social influence based scheme**. In this scheme, the top-$K$ nodes with the highest social influence invite others to form disjoint social clusters, and users can dynamically transfer across these clusters. Here, we set $K = 10$.

### B. Verification for Model Utility Function

In Figs. 4–7, we verify the model utility function in FL (measured by the loss of trained model) defined in (13) on MNIST dataset by training the 4-layer CNN model. Here, the number of participants (i.e., $N$) is set as 100. As shown in Figs. 4 and 5, the relationship between model loss and Gaussian noise scale under both IID and non-IID cases can be fitted as a sigmoid curve; while Fig. 6 shows that the relationship between model loss and non-IID degree can be fitted as an exponential function. In Fig. 7 and Fig. 8, we evaluate the loss of the trained model on two loss functions: typical negative log likelihood loss (NLLLoss) and the mean-square error (MSE), respectively. In Fig. 7, by varying both noise scale and non-IID degree, the model loss function can be well-fitted by the 3D sigmoid curve with curve fitting parameters $\mu_1 = 0.013$, $\mu_2 = 0.0044$, $\mu_3 = 0.0057$, $\mu_4 = 8.18$, and $\mu_5 = 0.14$. As seen in Fig. 8, the loss can still be well-fitted by 3D sigmoid curve, and the curve fitting parameters are $\mu_1 = 0.013$, $\mu_2 = 0.0021$, $\mu_3 = 0.0057$, $\mu_4 = 8.20$, and $\mu_5 = 0.14$. From Fig. 7 and Fig. 8, we can observe that the effect of LDP noise overwhelms the non-IID effect in model utility degradation. Besides, in Fig. 9, it can be observed that given $\theta_1 = 100$ and $\theta_2 = 1$, when $\alpha \in [0, \alpha_{th}]$, we have $\epsilon \in [0, 40]$; and when $\alpha \in [0.05, \alpha_{th}]$, we have $\sigma \in [0, 1.1]$. Under this setting, the customizable privacy budget $\epsilon$ is within a certain range, so that the added DP noise $\sigma$ will not destroy the model performance.

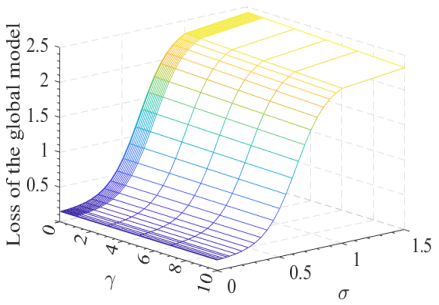



Fig. 7. Curve fitting of the loss function $\mathscr{L}(\sigma, \gamma)$ in (13) w.r.t noise scale $\sigma$ and non-IID degree $\gamma$ in MNIST using the NLLLoss loss function: $\mathscr{L}(\sigma, \gamma) = \frac{0.013\exp(-0.0044\gamma)}{0.0057+\exp(-8.18\sigma)} + 0.14$.

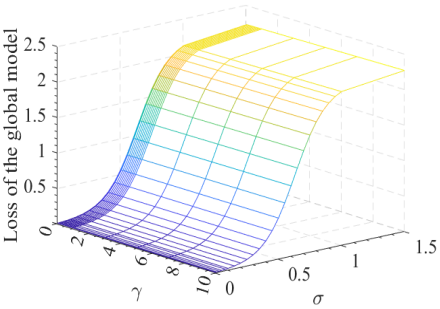



Fig. 8. Curve fitting of the loss function $\mathscr{L}(\sigma, \gamma)$ in (13) w.r.t noise scale $\sigma$ and non-IID degree $\gamma$ in MNIST using the MSE loss function: $\mathscr{L}(\sigma, \gamma) = \frac{0.013\exp(-0.0021\gamma)}{0.0057+\exp(-8.20\sigma)} + 0.14$.

### C. Numerical Results

Using Figs. 10–15, we first evaluate the effects of customized local perturbation, number of participants, and level of social effects in SCFL on MNIST and CIFAR-10 datasets. Next, we analyze the user payoff and stability of the federation game in SCFL using Figs. 16–19, by comparing with existing schemes. After that, we validate the feasibility of SCFL in complex models and language tasks using Figs. 20–21.

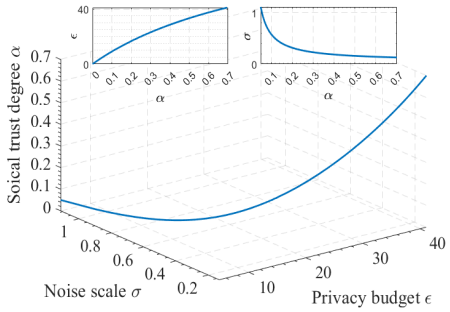



Fig. 9. The relationship between social trust degree $\alpha$, privacy budget $\epsilon$, and noise scale $\sigma$ in MNIST when $\theta_1 = 100$ and $\theta_2 = 1$.

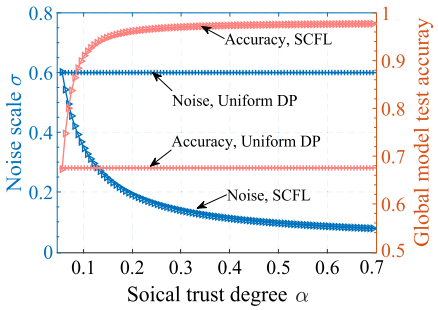



Fig. 10. Effect of customized local perturbation in terms of noise scale $\sigma$ and global model test accuracy vs. social trust degree in MNIST.

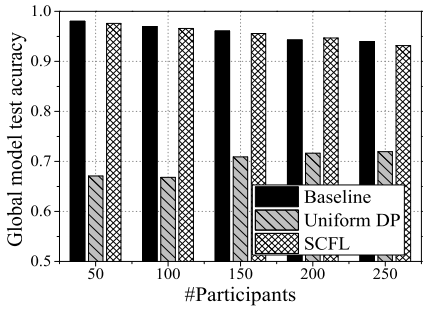



Fig. 11. Comparison of global model test accuracy in MNIST in three schemes under different numbers of participants.

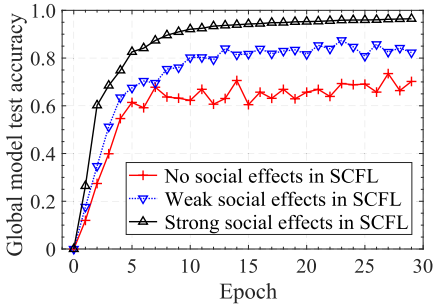



Fig. 12. Evolution of global model test accuracy in MNIST under different levels of social effects in SCFL ($N = 100$).

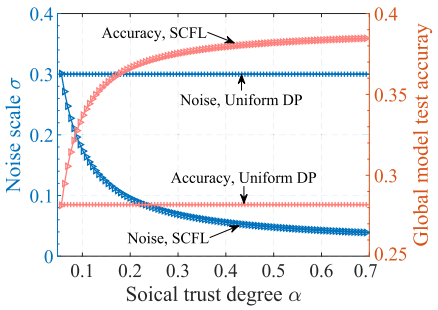



Fig. 13. Effect of customized local perturbation in terms of noise scale $\sigma$ and global model test accuracy vs. social trust degree in CIFAR-10.

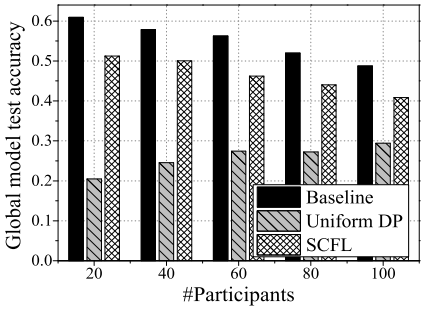



Fig. 14. Comparison of global model test accuracy in CIFAR-10 in three schemes under different numbers of participants.

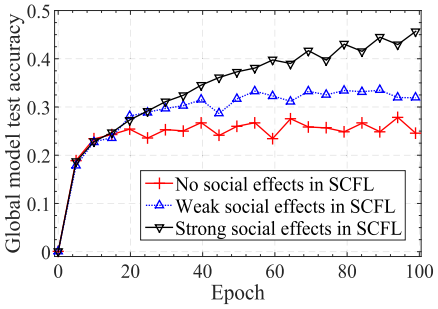



Fig. 15. Evolution of global model test accuracy in CIFAR-10 under different levels of social effects in SCFL ($N = 100$).

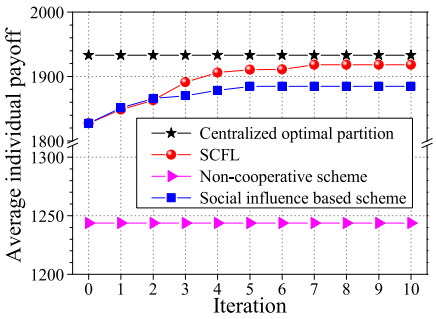



Fig. 16. Evolution of average individual payoff in SCFL, compared with other three conventional schemes ($N = 100$).

**Effect of Customized Local Perturbation.** Figs. 10 and 13 illustrate the effects of the average social trust degrees in terms of Gaussian noise scale and global model test accuracy on the MNIST dataset and CIFAR-10 dataset, respectively. As seen in Figs. 10 and 13, a higher social trust degree $\alpha$ in SCFL results in a lower scale of added Gaussian noise and correspondingly higher model accuracy on both MNIST and CIFAR-10, thereby enforcing customizable privacy protection. While in the uniform DP scheme, as the uniform and relatively large Gaussian noise is applied for all users, the global model accuracy keeps unvaried and at a low level under different $\alpha$.

**Effect of Number of Participants.** Figs. 11 and 14 compare the FL model performance in three schemes on MNIST and

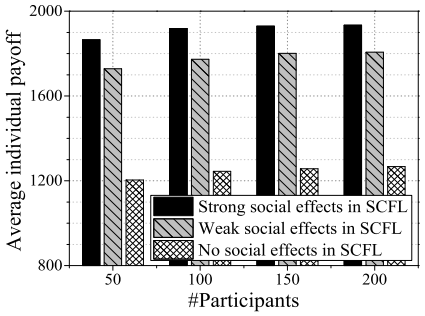



Fig. 17. Comparison of average individual payoff in SCFL under different numbers of participants and different levels of social effects.

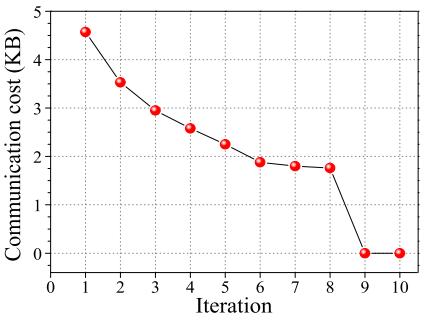



Fig. 18. Evolution of communication cost in forming a Nash-stable partition of social clusters in SCFL ($N = 100$).

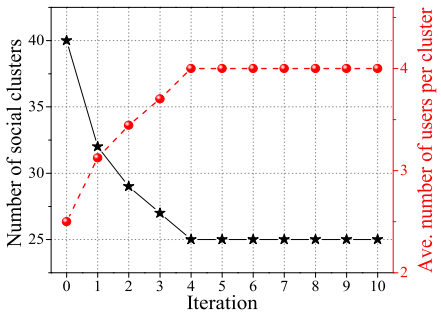



Fig. 19. Evaluation of social clustering results of the FTU game in terms of number of social clusters and average number of users per cluster ($N = 100$).

CIFAR-10, respectively. In the baseline scheme (i.e., naive FL under IID without DP perturbation), participants' raw local updates rather than the noised version are exchanged, where the social clustering, non-IID efects, and user privacy preservation are not considered. As shown in Figs. 11 and 14, the SCFL outperforms the uniform DP scheme in attaining a smaller accuracy gap with the baseline scheme in both MNIST and CIFAR-10 datasets. Besides, when the number of participants increases, the model accuracy in both the SCFL and the baseline scheme decreases, while that in the uniform DP scheme increases. The reason is that in our setting, the total dataset is divided among all participants in an non-IID manner. As such, more participants result in lower local samples of each user, causing an accuracy drop in our SCFL and the baseline scheme. Moreover, when more participants add the random noise with the same Gaussian distribution, the aggregated effect of LDP noise can be reduced, thereby causing an accuracy rise in the uniform DP scheme.

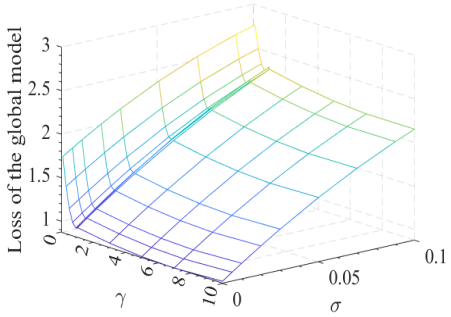



Fig. 20. Curve fitting of the loss $\mathscr{L}$ w.r.t noise scale $\sigma$ and non-IID degree $\gamma$ in the CIFAR10 dataset using the Resnet18 model.

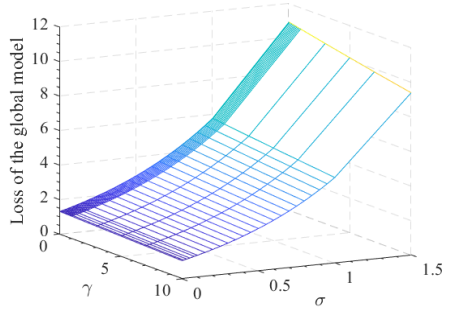



Fig. 21. Curve fitting of loss $\mathscr{L}$ w.r.t noise scale $\sigma$ and non-IID degree $\gamma$ in the 20newsgroup dataset using the MLP model.

**Level of Social Effects.** Figs. 12 and 15 depict the evolution of global model test accuracy in our SCFL under different levels of social effects in MNIST and CIFAR-10 datasets, respectively. As observed in Figs. 12 and 15, the test accuracy of global model in FL under strong social effects is higher than that under weak or no social effects on both MNIST and CIFAR-10, validating the effects of social attributes in combination with FL.

**Individual Payoff.** Fig. 16 compares the average individual payoff in our SCFL, compared with other three conventional schemes. In Fig. 16, it can be seen that the SCFL fast converges after 7 iterations and yields a near-optimal performance compared with the centralized optimal partition, which is superior to the non-cooperative scheme and social influence based scheme. It is because the partition structure with fixed cluster heads in the social influence based scheme cannot adapt to heterogeneous users and the varying FL environment, leading to a smaller individual payoff. In the non-cooperative scheme, as all users work alone in the FL process, it causes the lowest individual payoff. Fig. 17 shows the average individual payoff in our SCFL under different numbers of participants and various levels of social effects in FL. As seen in Fig. 17, users usually obtain better payoffs given more participants under strong or weak social effects, as a larger network size can increase the chance of seeking better cooperating partners. Besides, the higher social effect also results in better individual payoff, validating the potential of social ties in FL.

**Incurred Overheads.** Fig. 18 shows that the communication cost per iteration is in a decline and is less than 5 KB, during the formation of a Nash-stable partition among

$$\mathscr{L}(\sigma,\gamma)=\begin{cases}1.951-2.132\gamma+14.21\sigma+1.163\gamma^2+3.782\gamma\sigma-44.68\sigma^2, & \gamma\leq 1,\\ 1.026-0.042\gamma+16.83\sigma+0.003\gamma^2-0.0775\gamma\sigma-35.54\sigma^2, & \gamma>1.\end{cases} \tag{22}$$

100 participants. Besides, as a Nash-stable partition of social clusters is formed at 7 iterations, no user tends to transfer to other social clusters for improved individual payoff, thereby the communication cost drops to zero after 8 iterations. Additionally, the social cluster formation process for each FL task only occurs at the initial phase of FL learning. Thereby, our SCFL only incurs small additional overheads when the social cluster changes.

**Convergence and Stability.** Fig. 19 shows the social clustering results of our proposed FTU game in terms of the number of social clusters and average number of users per cluster. In Fig. 19, the network starts with a random clustering partition with 40 social clusters and converges to a final stable partition made up of 25 social clusters with an average of 4 users per cluster after 7 iterations, which accords with the empty core and Nash-stability of our FTU game in Theorems 2–3. Notably, during 4-7 iterations, there only exist inter-cluster member exchanges, i.e., no social users choose to form a singleton. As such, both the number of social clusters and average number of users per cluster remain unchanged during 4-7 iterations. Moreover, as seen in Figs. 16 and 19, the proposed Alg. 1 can quickly converge to the Nash-stable partitions within only 7 iterations under FL when $N = 100$.

**Discussions on Complex Models and Language Tasks.** For different FL tasks and models, the fitting curve of the loss may vary. However, the basic trend should remain the same, that is, with the increase of $\sigma$ or the decrease of $\gamma$, the accuracy of the model decreases and the loss increases. To validate this observation, we conduct additional experiments on the Resnet18 model for image classification tasks on CIFAR10 in Fig. 20 and the multilayer perceptron (MLP) model for text classification tasks in Fig. 21.

First, we utilize the Resnet18 model to perform similar experiments on CIFAR10 with learning rate of $5 \times 10^{-3}$, 30 participants, 30 global rounds, and local batch size of 64. As seen in Fig. 20, the loss cannot be fitted by 3D sigmoid curve, as the effects of $\gamma$ are more significant for the Resnet18 model on the CIFAR10 dataset than the CNN model on the MNIST dataset. Instead, we observe that if $\gamma \leq 1$, the effect of $\gamma$ is more obvious. When $\gamma > 1$, the effect of $\gamma$ is negligible and is masked by the effect of $\sigma$. The loss of the trained Resnet18 model can be fitted by a piecewise polynomial function, as in (22), shown at the top of the page.

Next, we train the MLP model using the 20newsgroup dataset [49] for text classification task under FL with 5 clients, 10 global rounds, learning rate of 0.001 and local batch size of 32. As depicted in Fig. 21, unlike image classification tasks, the 3D fitting formula exhibits an exponential trend, where the loss continues to increase even when the noise disrupts the training process. The fitting formula is $\mathscr{L} = 2.053\exp(-0.0139\gamma + 1.1574\sigma) - 0.7306$.

## VI. Conclusion

Striving a trade-off between privacy and utility plays a fundamental role in the practical deployment of DP-based FL services. Due to the rich manners of connectivity in real-world and motivations to ally for profits, we argue that the participants of FL tend to form social connections. By exploiting such social attributes among users, this paper has proposed a novel, efficient, and practical SCFL framework to realize feasible, high-utility, and customized privacy-preserving FL services. Firstly, we have designed a contribution quantification and fair allocation mechanism for heterogeneous users in each social cluster. Considering some clusters may have low mutual trust, a customizable privacy preservation mechanism has been devised for adaptive local model update sanitization based on social trust. In addition, we have developed a distributed two-sided matching algorithm to obtain an optimized stable partition in the FTU game. Experiment results have validated the effectiveness of SCFL in terms of user payoff, model utility, and privacy preservation, compared with existing solutions.

In future work, we will further investigate the social-aware personalized FL framework by forming social clusters for users with similar social interests and learning a personalized model within each social cluster.

## References

[1] Gartner. *What Edge Computing Means for Infrastructure and Operations Leaders*. Accessed: Feb. 10, 2022. [Online]. Available: https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders

[2] Y. Wang, Y. Pan, M. Yan, Z. Su, and T. H. Luan, "A survey on ChatGPT: AI—Generated contents, challenges, and solutions," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 280–302, 2023.

[3] P. Sun, G. Liao, X. Chen, and J. Huang, "A socially optimal data marketplace with differentially private federated learning," *IEEE/ACM Trans. Netw.*, 2024, doi: 10.1109/TNET.2024.3351864.

[4] Y. Wang, H. Peng, Z. Su, T. H. Luan, A. Benslimane, and Y. Wu, "A platform-free proof of federated learning consensus mechanism for sustainable blockchains," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3305–3324, Dec. 2022.

[5] A. Rodio, F. Faticanti, O. Marfoq, G. Neglia, and E. Leonardi, "Federated learning under heterogeneous and correlated client availability," *IEEE/ACM Trans. Netw.*, 2023, doi: 10.1109/TNET.2023.3324257.

[6] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Jun. 2021.

[7] Z. Su et al., "Secure and efficient federated learning for smart grid with edge-cloud collaboration," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1333–1344, Feb. 2022.

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 739–753.

[9] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333.

[10] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[11] W. Wei, L. Liu, Y. Wut, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 797–807.

[12] X. Shen, Y. Liu, and Z. Zhang, "Performance-enhanced federated learning with differential privacy for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24079–24094, Dec. 2022.

[13] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 170–185, Jan. 2021.

[14] M. S. E. Mohamed, W.-T. Chang, and R. Tandon, "Privacy amplification for federated learning via user sampling and wireless aggregation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3821–3835, Dec. 2021.

[15] T. Liu, B. Di, B. Wang, and L. Song, "Loss-privacy tradeoff in federated edge learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 546–558, Apr. 2022.

[16] K. Wei et al., "Low-latency federated learning over wireless channels with differential privacy," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 290–307, Jan. 2022.

[17] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[18] Y. Wang, Z. Su, and M. Yan, "Social metaverse: Challenges and solutions," *IEEE Internet Things Mag.*, vol. 6, no. 3, pp. 144–150, Sep. 2023.

[19] Y. Wang, Z. Su, T. H. Luan, R. Li, and K. Zhang, "Federated learning with fair incentives and robust aggregation for UAV-aided crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3179–3196, Sep. 2022.

[20] Y. Zeng, Y. Lin, Y. Yang, and J. Liu, "Differentially private federated temporal difference learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2714–2726, 2021.

[21] Z. Shi, G. Yang, X. Gong, S. He, and J. Chen, "Quality-aware incentive mechanisms under social influences in data crowdsourcing," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 176–189, Feb. 2022.

[22] X. Lin, J. Wu, J. Li, X. Zheng, and G. Li, "Friend-as-learner: Socially-driven trustworthy and efficient wireless federated edge learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 269–283, Jan. 2023.

[23] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "SEAR: Secure and efficient aggregation for Byzantine-robust federated learning," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 3329–3342, Oct. 2022.

[24] Y. Deng et al., "FAIR: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[25] X. Ma, X. Sun, Y. Wu, Z. Liu, X. Chen, and C. Dong, "Differentially private Byzantine-robust federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 3690–3701, Dec. 2022.

[26] C. Xu, Y. Jia, L. Zhu, C. Zhang, G. Jin, and K. Sharif, "TDFL: Truth discovery based Byzantine robust federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4835–4848, Dec. 2022.

[27] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "FLCert: Provably secure federated learning against poisoning attacks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3691–3705, 2022.

[28] Z. Li, H. Shen, and K. Sapra, "Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1745–1759, Sep. 2013.

[29] Y. Wang, Z. Su, and N. Zhang, "BSIS: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3620–3631, Jun. 2019.

[30] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.

[31] T. Halabi and M. Zulkernine, "Trust-based cooperative game model for secure collaboration in the Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[32] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 308–318.

[33] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Proc. Int. Conf. Adv. Neural Inf. Proces. Syst.*, vol. 31, Dec. 2018, pp. 6280–6290.

[34] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE INFOCOM*, 2020.

[35] B. Y. Lin et al., "FedNLP: Benchmarking federated learning methods for natural language processing tasks," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 157–175.

[36] T. M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," in *Proc. NeurIPS Workshop Federated Learn.*, vol. 26, 2019, pp. 1–5.

[37] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

[38] Q. Liu, S. Huang, J. Opadere, and T. Han, "An edge network orchestrator for mobile augmented reality," in *Proc. IEEE INFOCOM*, 2018, pp. 756–764.

[39] T. Minka, "Estimating a Dirichlet distribution," MIT, Cambridge, MA, USA, Tech. Rep., 2012, pp. 1–15. [Online]. Available: https://tminka.github.io/papers/dirichlet/minka-dirichlet.pdf

[40] J. Soch, *Proof: Maximum Likelihood Estimation for Dirichlet-Distributed Data*, The Book of Statistical Proofs, 2020, doi: 10.5281/zenodo.4305949. [Online]. Available: https://statproofbook.github.io/P/dir-mle.html

[41] L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. J. R. Liu, "Privacy or utility in data collection? A contract theoretic approach," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1256–1269, Oct. 2015.

[42] W. Saad, Z. Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 48–60, Jan. 2011.

[43] T. Wang, L. Song, Z. Han, and B. Jiao, "Dynamic popular content distribution in vehicular networks using coalition formation games," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 538–547, Sep. 2013.

[44] J. Leskovec and J. J. McAuley, "Learning to discover social circles in ego networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 539–547.

[45] Y. Wang et al., "Blockchain-based secure and cooperative private charging pile sharing services for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1857–1874, Feb. 2022.

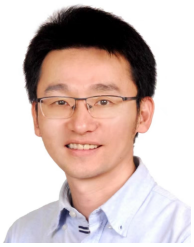
[46] J. Cao, T. Peng, Z. Qi, R. Duan, Y. Yuan, and W. Wang, "Interference management in ultradense networks: A user-centric coalition formation game approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5188–5202, Jun. 2018.

[47] S. Bera, S. Misra, and D. Chatterjee, "C2C: Community-based cooperative energy consumption in smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4262–4269, Sep. 2018.

[48] G. Liao, X. Chen, and J. Huang, "Social-aware privacy-preserving mechanism for correlated data," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1671–1683, Aug. 2020.

[49] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. ICML*, 1995, pp. 331–339.

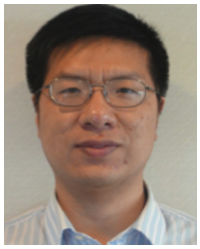
**Yuntao Wang** received the Ph.D. degree in cyberspace security from Xi'an Jiaotong University, Xi'an, China, in 2022. He is currently an Assistant Professor with the School of Cyber Science and Engineering, Xi'an Jiaotong University. His research interests include security and privacy in the intelligent IoT, network games, and blockchain.



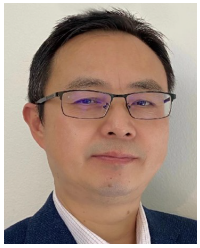
**Zhou Su** (Senior Member, IEEE) has published technical papers, including top journals and top conferences, such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. He received the Best Paper Award of International Conference IEEE ICC2020, IEEE BigdataSE2019, and IEEE CyberSciTech2017. He is an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY, and *IET Communications*.

**Yanghe Pan** is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include federated learning, data privacy in machine learning, and differential privacy.

**Tom H. Luan** (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, Canada, in 2012. He is currently a Professor with Xi'an Jiaotong University, China. He has authored/coauthored more than 97 journal articles and 58 technical papers in conference proceedings. His research interests include content distribution and media streaming in vehicular ad hoc networks and peer-to-peer networking and the protocol design and performance evaluation of wireless cloud computing and edge computing.



**Ruidong Li** (Senior Member, IEEE) received the D.Eng. degree from the University of Tsukuba in 2008. He is currently an Associate Professor with the College of Science and Engineering, Kanazawa University, Japan. His research interests include future networks, big data networking, blockchain, and network security. He is the Secretary of the IEEE ComSoC Internet Technical Committee and the Founder and the Chair of the IEEE SIG on Big Data Intelligent Networking and the IEEE SIG on Intelligent Internet Edge. He is the Guest Editor of prestigious journals, such as *IEEE Communications Magazine*, *IEEE Network Magazine*, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.



**Shui Yu** (Fellow, IEEE) received the Ph.D. degree in computer science from Deakin University, Australia, in 2004. He is currently a Professor with the School of Computer Science, University of Technology Sydney, Australia. He has published five monographs and edited two books, more than 500 technical articles at different venues, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. His research interests include cybersecurity, network science, big data, and mathematical modeling. He is an Elected Member of Board of Governors of IEEE VTS and IEEE ComSoc. He is also serving as the Editorial Boards for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Area Editor) and the IEEE INTERNET OF THINGS JOURNAL (Editor). He served as a Distinguished Lecturer for the IEEE Communications Society (2018–2021). He is a Distinguished Visitor of the IEEE Computer Society.