This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

Tackling Multiplayer Interaction for Federated Generative Adversarial Networks

Chuang Hu, *Member, IEEE,* Tianyu Tu, Yili Gong, *Member, IEEE,* Jiawei Jiang, *Member, IEEE,* Zhigao Zheng, *Member, IEEE,* and Dazhao Cheng, *Senior Member, IEEE*

Abstract—Generative Adversarial Networks (GANs) have become predominant in mobile computing for their ability to generate data. The concern for data privacy has made it arduous to collect large-scale datasets for GAN training on centralized servers. Federated Learning (FL) has emerged as a promising solution to address data privacy concerns. In this paper, we propose Oasis, a multiplayer-oriented federated GAN training system. We present a motivation, highlighting the Nash Equilibrium (NE) shift in vanilla federated GANs, exacerbated by data heterogeneity, leading to poor training performance with issues of vanishing gradient and mode collapse. To address mode collapse, Oasis extracts privacy-preserving data representations and generates a similarity table for clustering clients. Each group independently trains a GAN model and conducts distribution and fusion. By introducing a coordinator, Oasis generalizes intra-group games into *Separable Zero-sum Multiplayer Games* to tackle vanishing gradient. Thus, Oasis considers the overall federated GAN training as *Group-wise Separable Zero-sum Multiplayer Games*. Practically, we evaluate our theoretical results both on a hardware prototype and in a simulated environment. Evaluation results demonstrate the effectiveness of Oasis, with an average improvement of 23.13% and 26.33% in terms of FID and NDB/K respectively, compared to three *state-of-the-art* FL approaches over three datasets.

Index Terms—Generative Adversarial Networks (GANs), multiplayer zero-sum games, data heterogeneity.

1 Introduction

NOWADAYS, Generative Adversarial Networks (GANs) [1] are predominant generative models widely employed in mobile computing such as real-time indoor localization [2], edge image steganography [3], and smartphone malware detection [4]. A GAN consists of two networks: a discriminator and a generator. The generator aims to learn the distribution of real data, while the discriminator distinguishes between real data and data generated by the generator. These two components continuously compete against each other until convergence. This process can be understood as a zero-sum minimax game between two rational players, which differentiates GANs from other generative models, such as Stable Diffusion [5]. Training GANs requires a significant amount of data. Small datasets cause the discriminator to overfit to the training samples, making its feedback to the generator meaningless and causing the training process to diverge [6]. However, due to the increasing concern for data privacy and the enactment of relevant laws [7], collecting large-scale data and aggregating it on servers for training GANs have become challenging.

Federated Learning (FL) [8] has emerged as a prospective solution that facilitates distributed collaborative learning without disclosing original training data. In FL, data remains stored on the clients. During each round, the server selects a subset of clients and sends the model to them. These clients then train the models using their local data and upload the trained model updates back to the server. The server aggregates these updates using a technique called Federated Averaging (FedAvg) to tune the global model. The process of FL continues until the model convergence criteria are met.

However, currently a significant amount of research on FL is focused on optimizing supervised learning (SL) [9], [10] model training while neglecting the adaptation of unsupervised learning (UL) model training. Existing federated UL frameworks are largely based on federated analytics [11], achieving significant successes in areas such as privacy-preserving clustering of user interests for better advertisement recommendations [12], privacy-preserving collaborative frequent pattern mining [13], and privacy-preserving efficient communication node embedding [14]. In contrast, this work focuses on UL model training in a federated environment. Therefore, existing UL frameworks are not applicable to this work. Consequently, our discussion focuses on training SL and UL models. SL and UL have fundamental differences. SL can be described as learning the mapping from data to labels while UL focuses on recognizing the patterns of the data. Data heterogeneity (data is not independent and identically distributed (*i.i.d.*), non-*i.i.d.*) in FL results in varying levels of difficulty and focus when it comes to their adaptation. Data heterogeneity makes UL unable to capture data patterns, leading to the generation of meaningless results. GANs represent UL, and thus it is meaningful to research the integration of GANs into FL paradigms, *i.e.*, federated GANs.

Experts have researched this field. [15] modifies the weight allocation in FedAvg using Maximum Mean Discrepancy to address the impact of data heterogeneity on federated GANs. [16] finds that large-scale integration of GANs in FL paradigms leads to training inefficiency, which is then

- *Chuang Hu, Tianyu Tu, Yili Gong, Jiawei Jiang, Zhigao Zheng, Dazhao Cheng are with the Computer School, Wuhan University, Hubei 430072, China. E-mail: {handc, meetnailtu30, yiligong, jiawei.jiang, zhengzhigao, dcheng}@whu.edu.cn.*

addressed by utilizing balanced sampling and Kullback-Leibler (KL) weighting. While these studies have made some progress in addressing the challenges of integrating GANs into FL paradigms, they act as solutions based on statistical information during the training process, do not take experiments on large-scale clients into consideration, and do not tackle the root cause theoretically, *i.e.*, *the key distinction between two-player games and multiplayer games lies in the number of players involved and the computational complexity of interactions among them*. In this work, we approach the training of federated GANs from the perspective of multiplayer oriented GANs and treat them as *Group-wise Separable Zero-sum Multiplayer Games* (*GSZMGs*) to reach global optima. Mode collapse [17] occurs when the generator becomes stuck in a particular mode or pattern, failing to generate diverse outputs that cover the entire range of the data. Our findings indicate that the problem of mode collapse caused by data heterogeneity can be addressed by grouping GANs based on client data representations. Moreover, we propose the generalization of multiplayer zero-sum games to *Separable Zero-sum Multiplayer Games* (*SZMGs*) [18] within each group, thereby reducing the computational complexity of Nash Equilibrium (NE).

We propose a system called Oasis for ***O***rchestrating Gener***a***tive Adversarial Network***s*** in Federated Learn***i***ng Paradigm***s***. Oasis extracts privacy-preserving representations of client data (Section 4.1) and generates a similarity table based on the extracted data representations (Section 4.2). It is *challenging* to extract data representations for clustering while preserving clients' privacy. Clients are then grouped based on the similarity table and each group trains a GAN model pair independently. Model distribution and fusion are performed at the group level (Section 4.3). Simply adapting GANs to FL paradigms results in *Pair-wise Zero-sum Polymatrix Games* (*PZPGs*) [18] within groups, which are PPAD-complete [19]. To reduce the computational complexity of multiplayer zero-sum games, Oasis introduces a coordinator to simplify the games into *SZMGs*, whose representing matrices are of low rank (Section 4.4). Designing a common coordinator for different *SZMGs* is a *challenge*. In Oasis, the overall training corresponds to *GSZMGs*, which are designed to optimize towards global optima.

We evaluate Oasis both on a hardware prototype with 20 Raspberry Pi devices and in a simulated environment. The evaluation conducted demonstrates that Oasis outperforms three other *state-of-the-art* FL approaches, achieving an average improvement of 23.13% and 26.33% in terms of fréchet inception distance (FID) [20] and number of statistically-different bins (NDB) divided by K (NDB/K) [21] respectively. Furthermore, a series of detailed analyses provide compelling evidence for the effectiveness of Oasis.

In summary, this work contributes the following:

1) We make a case for federated GANs. We observe the presence of NE shift issue in federated GANs, along with the occurrence of vanishing gradient and mode collapse problems in non-*i.i.d.* settings, which are caused by the interactions of multiple players in the game and data heterogeneity, respectively.
2) We design Oasis, a system that considers federated GANs as *GSZMGs*. Oasis extracts privacy-preserving data representations and utilizes them as clustering metrics. Based on clustering results, Oasis trains GANs by group. Additionally, Oasis integrates a regularizer into the local GANs, aiming to achieve global optima. Furthermore, we demonstrate the convergence of Oasis.
3) We evaluate Oasis with three different datasets and their corresponding GANs, both on a hardware prototype with 20 Raspberry Pi devices and in a simulated environment, compared to three *state-of-the-art* FL approaches. Evaluation results verify *GSZMGs* achieve global optima and demonstrate that Oasis enhances the data generation capabilities of federated GANs, achieving an average improvement of 23.13% on FID and 26.33% on NDB/K.

2 Background and Motivation

2.1 GANs and Two-player Zero-sum Games

The essence of GANs is to minimize the Jensen-Shannon divergence between real data distribution $\mathbb{P}$ and fake data distribution learned by the model $\mathbb{Q}$. We utilize the general f-divergence GAN (f-GAN) [22], [23] objective to formulate GANs, which are induced by convex functions f^c:

$$F(\mathbb{P}, \mathbb{Q}; \psi) \equiv \mathbf{E}_{\mathbb{P}}[\psi] - \mathbf{E}_{\mathbb{Q}}[f^c \circ \psi], \tag{1}$$

where ψ denotes the discriminator.

In the context of GANs, the adversarial training between the generator and discriminator is interpreted as a zero-sum game between two players. The convergence situation of training for both the generator and discriminator corresponds to the state of NE in the zero-sum game between these two players. Formally, NE is defined as follows [24]:

Definition 1. *(Nash Equilibrium) Let S_i be the all possible strategy set for one player i out of N players, where $i = 1, ..., N$. Let $s^* = (s_i^*, s_{-i}^*)$ denote a strategy profile set consisting of one strategy for each player, where s_{-i}^* denotes the $N-1$ strategies of all the players except player i. Let $u_i(s_i, s_{-i}^*)$ denote the utility function of player i's strategies. The strategy profile s^* is a Nash Equilibrium (NE) if $\forall s_i \in S_i, u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$.*

The NE of two-player zero-sum game is a situation where player p's payoff $u_p(s_p^*, s_{-p}^*)$ is equivalent to player q's loss, resulting in the zero net improvement in benefit of the game, *i.e.*, $u_p(s_p^*, s_{-p}^*) = -u_q(s_q^*, s_{-q}^*)$.

The strategy of the generator player is to maximize the utility function (*i.e.*, minimize the loss function) to increase the overall payoff, while the strategy of the discriminator player is to minimize the utility function (*i.e.*, increase the loss function) leading to an overall decrease in payoff.

Training GANs has two notorious issues: mode collapse and vanishing gradient [17]. Mode collapse refers to a situation where the generator only learns a subset of the data distribution, failing to generate diverse data. Vanishing gradient occurs when the discriminator is trained too successfully, and the generator cannot obtain meaningful feedback from the discriminator, resulting in a persistent lack of gradient descent.

We propose the *loss discrepancy* (*ΔLoss*) between discriminator and generator as a metric for NE shift.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

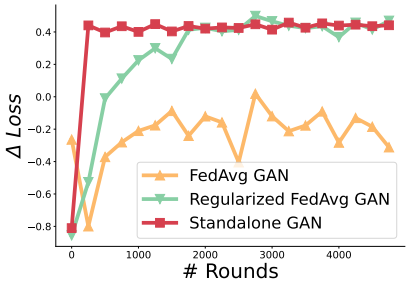


Fig. 1. Loss discrepancy (*i.i.d.*).

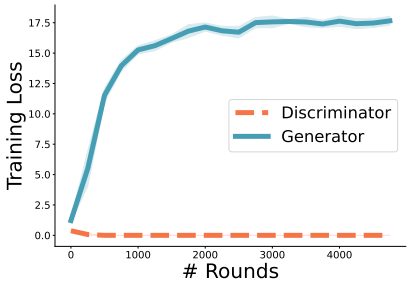


Fig. 2. Vanishing gradient (non-*i.i.d.*).

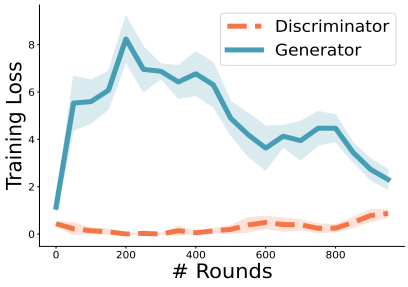


Fig. 3. *Regularization* (non-*i.i.d.*).

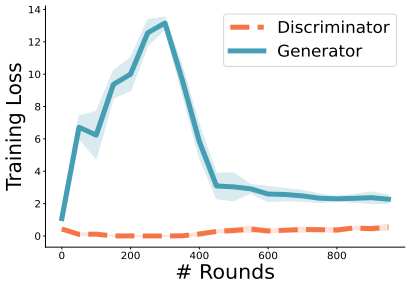


Fig. 4. *Clustering* (non-*i.i.d.*).

Fig. 5. Generated data by FedAvg GAN, regularized FedAvg GAN, and clustered FedAvg GAN respectively in non-*i.i.d.* settings.

Definition 2. *(Loss Discrepancy, ΔLoss) Let $\mathcal{D}$ be the loss of the discriminator, and $\mathcal{G}$ be the loss of the generator, loss discrepancy $\Delta Loss = \mathcal{D} - \mathcal{G}$.*

Upon reaching NE, no player can enhance her payoff by modifying her individual strategy, meaning that each player's payoff is maximized. In the context of GANs, the discriminator aims to maximize the loss function, while the generator seeks to minimize it. Therefore, the disparity between these two losses reflects the overall payoff for the players, with a larger discrepancy indicating closer proximity to NE. Intuitively, we utilize the loss discrepancy of standalone GANs to assess whether federated GANs have reached NE. A larger disparity in loss discrepancy compared to standalone GANs indicates a greater deviation from NE.

We employ the FID to measure the generated quality and NDB/K to measure the generated diversity. Both evaluation metrics aim for a *lower* value, indicating *better* performance. For more detailed information, please refer to Section 7.1.

2.2 Federated GANs and Multiplayer Zero-sum Games

This paper considers a federated GAN scenario that each client is equipped with a GAN model, consisting of a generator and a discriminator while the server, on the other hand, does not possess an actual GAN model, and its role is solely to aggregate the models from the client side. Specifically, the objective function of federated GANs is

$$\begin{aligned} \min_{\mathbb{Q}} \max_{\psi} f(\mathbb{P}, \mathbb{Q}; \psi) &= \sum\nolimits_{k=1}^{N} p_k F_k(\mathbb{P}, \mathbb{Q}; \psi) \\ &= \mathbb{E}_k[F_k(\mathbb{P}, \mathbb{Q}; \psi)], \end{aligned} \tag{2}$$

where N is the number of clients, $p_k \geq 0$, and $\sum_k p_k = 1$. Generally, we set $p_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ with n_k samples available at each client k.

Expanding a two-player game to a multiplayer setting results in a *Graphical Polymatrix Game* (*GPG*). Formally [18],

Definition 3. *(Graphical Polymatrix Games) A Graphical Polymatrix Game (GPG) is defined in terms of an undirected graph $G = (V, E)$, where V is the set of players of the game and every edge $(u, v) \in E$ is associated with a two-player game between its endpoints.*

If every edge of a *GPG* is related to a *zero-sum* game between its endpoints, then the game is defined as a *Pairwise Zero-sum Polymatrix Game* (*PZPG*), where total payoff across all players remains *zero-sum*. A *Separable Zero-sum Multiplayer Game* (*SZMG*) is that players pursue global zero-sum objectives rather than local zero-sum objectives. Formally [18],

Definition 4. *(Separable Zero-Sum Multiplayer Games) A Separable Zero-sum Multiplayer Game (SZMG) $\mathcal{GG}$ is a GPG in which, for any pure strategy profile f, the sum of all players' payoffs U is zero, i.e., $\forall f, \sum_{u \in U} \mathcal{P}_u(f) = 0$.*

The two-player games in *SZMGs* are defined as lawyer games:

Definition 5. *(Lawyer Games) $\forall (u, v) \in E$, let $\mathcal{GG} := \{A^{u,v}, A^{v,u}\}$ be an n-player SZMG, such that every player $u \in [n]$ has m_u strategies and set $A^{u,v} = A^{v,u} = 0$ for all pairs $(u, v) \notin E$. Lawyer Games are the pairs $(u, v) \in E$.*

Considering each local player pair lacks knowledge about the strategies employed by other local player pairs, it becomes challenging for players to make optimal strategies. Therefore, we propose *Group-wise Separable Zero-sum Multiplayer Games* (*GSZMGs*), which leverage external information (*e.g.*, the similarity in strategy choices among players in the same group) to optimize players' strategies. Formally,

Definition 6. *(Group-wise Separable Zero-Sum Multiplayer Games) Games $\mathcal{G}$ where players are partitioned into groups $\mathcal{G} := \{\mathcal{C}_1, ..., \mathcal{C}_K\}$ such that within a group $\mathcal{C}_i = (V_i, E_i)$, edges among competitors are SZMGs while edges among counterparts*

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

are coordination games are called Group-wise Separable Zero-sum Multiplayer Games (GSZMGs).

GSZMGs guides players not to pursue individual payoff maximization; it is a game oriented towards global NE, *i.e.*, maximizing the overall utility payoff of all players.

Table 1 shows the meaning of the main notations in the paper.

TABLE 1
Main notations in the paper.

Notation	Meaning
n	The number of clients
$\mathbb{P}$	Real data distribution
$\mathbb{Q}$	Fake data distribution
ψ	The discriminator
F_γ	Loss function of the discriminator
$F_\mathbb{Q}$	Loss function of the generator
$\mathcal{E}$	The data representation extraction function
$a_j^{(i)}$	The activation in Self-taught Learning
b_j	The base in Self-taught Learning
ζ_i	The silhouette score
h_i	The inter cluster distance
q_i	The intra cluster distance
$\mathbb{C}^*$	The similarity table
S_i	The strategy of a player i
u_i	The utility function of a player i
$\mathcal{G} =< V, E >$	A game $\mathcal{G}$ consisting of player set V and player relationship E
$A^{u,v}$	The utility matrix

2.3 Motivation

2.3.1 Towards Conceptualizing Federated GANs as SZMGs

Observation 1. *Multiplayer interactions of federated GANs contribute to NE shift in i.i.d. settings, impacting the* ***quality*** *of generated samples.*

As discussed in previous works [1], [25], GANs correspond to two-player zero-sum minimax games, namely the generator and the discriminator, which are self-interested agents that aim to reduce the opponent's payoff in order to maximize their own payoff. The shift of NE exists in federated GANs with *i.i.d.* data compared to the loss discrepancy of standalone GANs, as shown in Fig. 1. Federated GANs align with *PZPGs*, because generator and discriminator make independent decisions and mutually influence their payoffs within each client, represented by solid black lines in Fig. 7(a). During the federated process, all generators interact with each other, represented by dashed blue lines in Fig. 7(a), and the same applies to the discriminators, represented by dotted red lines in Fig. 7(a). The overall payoff remains constant since the interactions within each client are zero-sum.

Lemma 1. *Finding a NE in PZPGs with strictly competitive games on their edges is PPAD-complete [19].*

Theorem 1. *The aggregation of strategies from each local player results in a global strategy that leads to a deviation from the optima of the global NE, i.e., NE shift.*

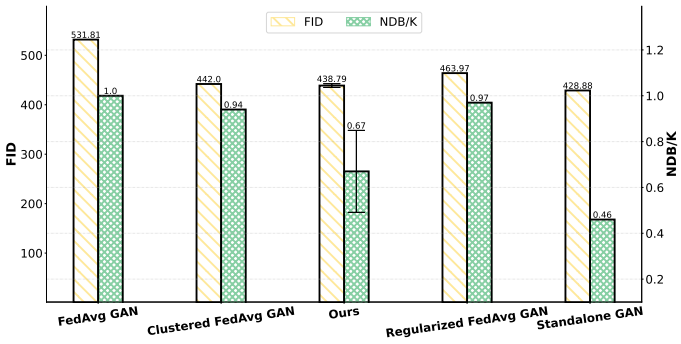
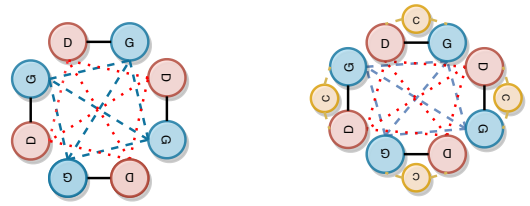


Fig. 6. Synthesis performance comparison (non-*i.i.d.* setting).



(a) *PZPGs* (b) *SZMGs*

Fig. 7. In the context of $four$-player-pair zero-sum multiplayer games, the interactions between the discriminator (D), generator (G), and coordinator (C) are represented by connected line segments between their respective symbols.

Proof. The f-divergence family f^c is always convex [22], and thus for each player i, we have $\mathbb{E}(u(s_i^*, s_{-i}^*)) \geq u(\mathbb{E}(s_i^*), s_{-i}^*)$ due to the property of convex functions, so that aggregating strategies makes global strategy away from the optima. □

Lemma 1 indicates that extending GANs to a federated context results in prolonged training convergence times. Theorem 1 highlights that the conventional approach of averaging local gradients in the federated context hinders the convergence of training for GANs. An intuitive remedy is to generalize *PZPGs* and reduce the self-interest of players, achieving global optima and reducing the computational complexity of NE. This is achieved by introducing a coordinator, ensuring that the interactions within each client are no longer zero-sum but maintaining a global zero-sum outcome. This concept aligns with the notion of *SZMGs* (see Fig. 7(b)), where for every local player pair $< p_i, q_i >$, there is a coordinator ϵ, so that $u_i(p_i) \circ \epsilon = -u_i(q_i) \circ \epsilon, s.t. \sum_i u_i(p_i) = -\sum_i u_i(q_i)$. [26] further demonstrates that matrices representing *SZMGs* are of low rank, implying the feasibility of solving such games. [18] obtains that *PZPGs* and *SZMGs* are payoff preserving transformation equivalent. Therefore, it is reasonable to generalize the games. We claim that the role of ϵ in *SZMGs* is analogous to the role of regularizer in GANs, and thus we construct federated GANs as *SZMGs* via *Regularization*. Experiments demonstrate *Regularization* addresses NE shift with *i.i.d.* data, as shown in the green line of Fig. 1.

Challenge 1. *GANs have various f-divergence implementations. Designing a regularization module for adapting different f-divergences is an obstacle.*

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

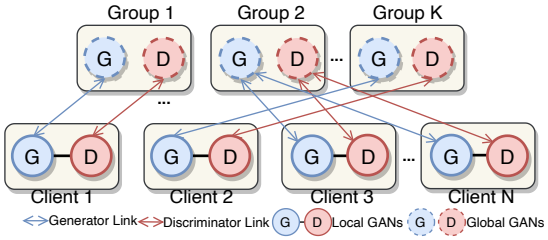


Fig. 8. The Oasis system topology.

2.3.2 Data Heterogeneity Brings About GSZMGs

Observation 2. *Data heterogeneity exacerbates NE shift, leading to mode collapse, and thus reducing the* ***diversity*** *of generated samples.*

When training federated GANs with non-*i.i.d.* data, NE shift becomes severer, accompanied by vanishing gradient, as depicted in Fig. 2, where the lines show the mean client loss, and the shaded regions represent the loss standard deviation (std), with the subsequent training loss graphs following the same representation. The left column of Fig. 5 exhibits mode collapse of the generated data. While *Regularization* addresses vanishing gradient, mode collapse persists, as shown in Fig. 3 and the middle column of Fig. 5 respectively. Therefore, it is necessary to introduce a method to address mode collapse caused by data heterogeneity.

GANs represent UL models. Unlike SL, which relies on labeled data for training predictive or classification models, UL explores unlabeled data to discover hidden patterns.

Theorem 2. *In contrast to SL, data heterogeneity further exacerbates the training difficulty of federated UL, making it challenging for UL to capture global data patterns.*

Proof. The task of SL is to learn a *data(X)-to-target(Y)* map $f : X \rightarrow Y$, while the task of UL is to get a *data-to-pattern* functional $f \circ g \rightarrow g(f)$. In FL, data heterogeneity mathematically means that there are node i and node j, for their local data X_i and X_j, $supp(X_i) \cap supp(X_j) \rightarrow \emptyset$. For SL, there is a gap between $\mathbb{E}(X) \rightarrow \mathbb{E}(Y)$ and optimal target Y^*. However, $supp(\mathbb{E}(Y)) \cap supp(Y^*) \neq \emptyset$ for target Y is of the same dimension. For UL, global averaging means that $\frac{\int_{X_i}\int_{X_j} g(f)\prod_x df(x)}{\sum_i X_i}$, and thus we cannot compute results for functions are not over a continuous range. Consequently, we conclude that SL gets deviated but meaningful results while UL cannot capture data patterns. □

Theorem 2 demonstrates that data heterogeneity contributes to mode collapse in federated GANs. Varying data distributions among clients render the aggregation of local model distribution $\mathbb{Q}$ ineffective. For example, one client learns cat data distributions, and another learns dog data distributions. Averaging them is ineffective because organisms that are half cat and half dog do not exist.

Inherently, resolving mode collapse in federated GANs necessitates techniques to *handle disparate data sources* while *preserving intrinsic patterns* within each client's data, which can be deduced as $\frac{\sum_{\mathcal{C}_k}\int\int_{(X_i,X_j)\in\mathcal{C}_k} g(f)\prod_x df(x)}{\sum_i X_i}$, where $\forall\, client\ data\ (X_i, X_j) \in cluster\ \mathcal{C}_k, supp(X_i) \cap supp(X_j) \neq \emptyset$. Therefore, we address the issue of data heterogeneity using *Clustering*. We conduct an experiment where we aim to cluster clients based on data classes and corresponding quantities uploaded. Experimental results demonstrate the effectiveness of clustering in alleviating mode collapse to some extent, as presented in Fig. 4 and the right column of Fig. 5.

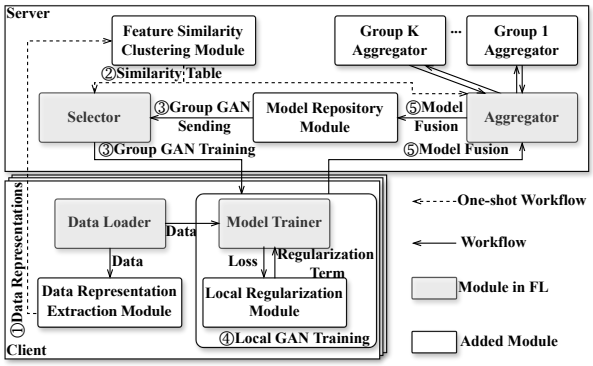


Fig. 9. The Oasis system architecture. The workflow of Oasis involves 1) Extracting data representations with privacy preservation and then 2) Constructing a Similarity Table. A global training round can be outlined as follows: 3) Group GAN Sending, 4) Local GAN Training, and 5) Model Fusion. Note that Step 1) and 2) are ***one-shot*** workflow.

Accordingly, we propose training federated GANs as *GSZMGs*, where players play *SZMGs* within each group. Fig. 6 illustrates the effects of *Regularization* and *Clustering* with non-*i.i.d.* data. Our proposed method combines both approaches, resulting in more pronounced improvements.

Challenge 2. *It is not feasible to directly have clients' data information in FL. Therefore, a major challenge is how to extract privacy-preserving data representations as cluster metrics and cluster reasonably.*

3 System Overview

To address the aforementioned two challenges, we present Oasis, a federated GAN training system, aiming to tackle issues arising from multiplayer interactions and data heterogeneity.

3.1 System Model

In Oasis, each client is equipped with a pair of generator and discriminator for training purposes. The server maintains certain global GANs responsible for fusing and distributing the models. Oasis organizes clients into groups, resulting in k global GANs on the server, where k denotes the number of groups. Throughout the FL process, a global model pair on the server corresponds to the local model pairs of clients in a grouped manner, forming the topology shown in Fig. 8.

3.2 Modular Design

Oasis (see Fig. 9) has a **Data Representation Extraction Module** (**DRE**) to obtain data distribution properties in a privacy-preserving manner. The core of Oasis is a **Feature**

Similarity Clustering Module (**FSC**), which employs clustering methods to group clients based on extracted representations. A similarity table is generated, which captures clients' representation similarity, guiding model fusion and distribution processes. **Model Repository Module** (**MRM**) partitions and archives selected clients' models based on the **FSC**-generated similarity table. Selected clients upload their locally trained models to the server. The server selectively groups and fuses the models using the FedAvg algorithm. To tackle multiplayer interaction issue in federated GANs caused by self-interest in zero-sum minimax games, Oasis introduces a **Local Regularization Module** (**LRM**), which incorporates a regularizer into the local GANs to play *SZMGs* within each group, aiming to reduce computational complexity of NE, and thus achieve global optima.

We emphasize that different modules are executed at different ends: **DRE** and **LRM** are located on the client end, while **FSC** and **MRM** are implemented on the server end.

The workflow of Oasis is described as follows: 1) Data Representation Extraction: **DRE** extracts data representations from clients, which are uploaded to the server. 2) Similarity Table Construction: **FSC** constructs a similarity table using representations. 3) Group GAN Sending: the server randomly selects clients, assigns groups based on the similarity table, and then sends global model pairs (generator and discriminator) from **MRM** to the clients by group. 4) Local GAN Training: the local GAN is trained via **LRM** for several iterations over the client's data, which is then uploaded to the server. 5) Model Fusion: **MRM** archives outcomes of fusion on the grouped uploaded model pairs according to the similarity table. 6) repeat Step 3) to 5) until global model pairs converge.

Player Interaction Analysis. During the training process, players interact within their respective groups, and players from different groups do not interact. We claim that the non-associated ***one-shot*** clustering among groups during the training process is suitable for federated GANs since each group learns some specific parts of the data distribution and data representation-based grouping ensures that each data distribution is learned. After training, combining the generator models from all groups provides the overall data distribution.

4 Oasis Design

In this section, we delve into the details of implementing the four modules. In **DRE**, our focus lies in discussing how to extract client data representations with privacy preservation. In **FSC**, we utilize the extracted representations for similarity clustering, taking into consideration both clustering quality and quantity. Moving on to **MRM**, we outline the global grouping fusion of generators and discriminators. Additionally, we provide evidence for the convergence bound in the training of Oasis. Finally, in **LRM**, we discuss the implementation of regularization (*i.e.*, coordination) for each local GAN, ensuring that the overall game is a *GSZMG*.

4.1 Data Representation Extraction

Due to the requirements of FL paradigms, data cannot leave the client. However, *GSZMGs* need to be formed within groups. Existing clustering metrics for federated clustering include model weights [27], [28] and data distribution [29], [30]. As observed in Section 2.3, federated GANs face the issue of vanishing gradient. Therefore, the former metric is not suitable for federated GANs and we group clients based on data distribution. Formally, we define how to extract user data distribution in a privacy-preserving manner as follows:

Problem 1. *(Opt-$\mathcal{E}$) Let $\mathcal{E}(x)$ be the representations of extraction function $\mathcal{E}$ for data x. A $\mathcal{E}^*$ is required so that $\forall \mathcal{F}, x \neq \mathcal{F}(\mathcal{E}^*(x))$, and thus $\mathcal{E}^*$ is privacy-preserving.*

We leverage Self-taught Learning (STL) [31] to deal with *Opt-$\mathcal{E}$* for its ability to extract data representations without labels. We apply STL to represent each client's local data as a sparse weighted combination of bases, capturing higher-level structure and distribution. STL exploits the uniqueness of sparse coding obtained from specific inputs as a distinctive feature to distinguish different data distributions among clients, thus extracting it as the data representation. Specifically, applying STL to Oasis involves the following two steps. 1) Given the local unlabeled data $\{x_u^{(1)}, ..., x_u^{(k)}\}$ with each $x_u^{(i)\in R^n}$, STL solves the following optimization problem:

$$\begin{aligned} \min_{b,a} \quad & \textstyle\sum_i \|x_u^{(i)} - \sum_j a_j^{(i)} b_j\|_2^2 + \beta \|a^{(i)}\|_1, \\ s.t. \quad & \|b_j\|_2 \leq 1, \forall j \in 1, ..., s. \end{aligned} \tag{3}$$

$a_j^{(i)}$ is the activation of base b_j for input $x_u^{(i)}$. Sparse coding is applied to the unlabeled data $x_u^{(i)} \in R^s$ to learn a set of bases b. 2) With learned b, for each local data $x_l^{(i)} \in R^s$, $\hat{a}(x_l^{(i)}) \in R^s$ is computed by solving the following optimization problem:

$$\hat{a}(x_l^{(i)}) = \arg\min_{a^{(i)}} \|x_l^{(i)} - \textstyle\sum_j a_j^{(i)} b_j\|_2^2 + \beta \|a^{(i)}\|_1, \tag{4}$$

where the sparse vector $\hat{a}(x_l^{(i)})$ is the privacy-preserving representation for $x_l^{(i)}$.

Theorem 3. *$\hat{a}$ is the object $\mathcal{E}^*$, and thus the STL-based representation extraction is privacy-preserving.*

Proof. The representation $\hat{a}(x_l^{(i)})$ is uploaded to the server while bases b are kept in the client. We take the derivative of Eq. 4 to obtain the minimum value. For convenience, we compute $\mathcal{E} = (x_l^{(i)} - \sum_j a_j^{(i)} b_j)^2 + \beta (a^{(i)})^2$. $\mathcal{E}' = 2(x_l^{(i)} - \sum_j a_j^{(i)} b_j)(-\sum_j b_j) + 2\beta(a_j^{(i)}) = 0$, which is simplified to $\hat{a} = \sum_j a_j^{(i)} = \frac{\sum_j b_j x_l^{(i)}}{(\sum_j b_j)^2 + 1}$. To recover $x_l^{(i)} = \frac{\hat{a}((\sum_j b_j)^2+1)}{\sum_j b_j}$, both $\hat{a}$ and b are required, *i.e.*, without b, $x_l^{(i)}$ cannot be recovered. Thus, $\hat{a}$ is the privacy-preserving $\mathcal{E}^*$. □

Through the proof of Theorem 3, we successfully address Problem 1 by identifying a STL-based method to extract data representations with privacy preservation.

4.2 Feature Similarity Clustering

We group clients based on the extracted data representations. During the training process, client data remains unchanged, so that ***one-shot*** clustering is sufficient to group

Algorithm 1: *Quality-Quantity* Trade-off *K-means*

Input: Representations $\mathcal{R}$, quantity threshold of cluster data η, number of clients m
Output: Similarity table $\mathbb{C}^*$

1 Initialize optimal score $\zeta^* \leftarrow -1$; optimal clusters $\mathbb{C}^* \leftarrow \emptyset$;
2 **for** $k = 1 : M$ **do**
3 Get cluster $\mathbb{C}$ based on *K-means*;
4 Get silhouette score ζ^k based on Eq.5;
5 **if** $\zeta^* < \zeta^k$ **then**
6 $\zeta^* \leftarrow \zeta^k$; $\mathbb{C}^* \leftarrow \mathbb{C}$;
7 **end**
8 **end**
9 **for** $\mathbb{C}_i \subset \mathbb{C}^*$ **do**
10 **while** $|\mathbb{C}_i| < \eta$ **do**
11 distance to the nearest cluster $d^* \leftarrow +\infty$;
12 **for** $\mathbb{C}_j \subset (\mathbb{C}^* \setminus \mathbb{C}_i)$ **do**
13 distance to $\mathbb{C}_j$ $d \leftarrow \sum_{x \in \mathbb{C}_i} \sum_{y \in \mathbb{C}_j} \|x - y\|$;
14 **if** $d^* > d$ **then**
15 $d^* \leftarrow d$;
16 the nearest cluster $\mathbb{C}_n \leftarrow \mathbb{C}_j$;
17 **end**
18 **end**
19 $\mathbb{C}_i \leftarrow \mathbb{C}_i \cup \mathbb{C}_n$; $\mathbb{C}^* \leftarrow \mathbb{C}^* \setminus \mathbb{C}_n$;
20 **end**
21 **end**

clients. We consider extracted representations as points in a high-dimensional space and employ *K-means* algorithm [32] for ***one shot*** clustering analysis, which is widely applied in approximate nearest neighbor search [33] for its low time complexity. Due to the inability to guarantee the quality of extracted data representations (*i.e.*, data is non-spherical), the clustering results may be suboptimal, and a fixed configuration of the cluster number is often inefficient. We define the problem as:

Problem 2. *(Opt-$\mathbb{C}$) Let m be the number of clients. Given K-means fails to determine the optimal cluster number k ($1 \leq k \leq M$) with data representations $\mathcal{R}$, how can Oasis obtain the optimal clusters $\mathbb{C}^*$?*

To address *Opt-$\mathbb{C}$*, we propose Algorithm 1. We utilize silhouette scores [34] to evaluate the quality of each cluster, which for a data point i is given as

$$\zeta_i = \frac{h_i - q_i}{max(h_i, q_i)}, \tag{5}$$

where h_i is the inter cluster distance defined as the average distance to the nearest non-membership cluster of i

$$h_i = \min_{k \neq i} \frac{1}{|\mathbb{C}_k|} \sum\nolimits_{j \in \mathbb{C}_k} d(i,j), \tag{6}$$

and q_i is the intra cluster distance, defined as the average distance to all other points within the cluster

$$q_i = \frac{1}{|\mathbb{C}_i| - 1} \sum\nolimits_{j \in \mathbb{C}_i, i \neq j} d(i,j). \tag{7}$$

To obtain the similarity table, we firstly perform clustering and calculate ζ^k for each k (line 2-7). The optimal number of clusters $k^* = \arg\max_k \zeta^k$. This approach objectively determines the appropriate number of clusters based on clustering result quality. However, obtaining high-quality clustering doesn't guarantee sufficient data quantity within each cluster for model training, so that we merge clusters with their nearest neighbors if the data quantity falls below a predefined threshold η, iterating until it meets the threshold (line 10-19). Therefore, we have similarity table $\mathbb{C}^* = \mathbb{C}(\zeta, \eta)$, where threshold η controls the trade-off between quality and quantity. We conduct further ablation evaluation on η in Section 7.4.

Complexity Analysis. It takes $\mathcal{O}(ntk)$ time to solve *K-means* algorithm and $\mathcal{O}(n^2)$ time to obtain the silhouette score, where n represents the number of data points and t represents iteration times. After searching for each possible k, we obtain $m(\mathcal{O}(ntk) + \mathcal{O}(n^2)) \approx \mathcal{O}(m(n^2))$. Checking the quantity threshold and updating clusters require $\mathcal{O}(n^2)$ time. Therefore, we conclude that the ***one-shot*** clustering algorithm costs $\mathcal{O}(m(n^2)) + \mathcal{O}(n^2) \approx \mathcal{O}(m(n^2))$ time.

4.3 Model Repository Module

Models of clients are partitioned into clusters based on the similarity table $\mathbb{C}^*$ after local training. This partitioning aims to tailor federated GAN models collaboratively. Each cluster represents a specific global model pair, *i.e.*, through this module, we transform *SZMGs* into *GSZMGs*, making an environment for within-group zero-sum game. *GSZMGs* achieve global optima, leading to global NE, and thus under *GSZMGs*, federated GANs converge to some constant.

In j_{th} global round, for selected clients $\mathcal{C}_i^j$ belonging to group $\mathcal{C}_i$, we have

$$< \mathbb{Q}_i, \psi_i > = \sum\nolimits_{k=1}^{|\mathcal{C}_i^j|} \frac{n_k}{n} < \mathbb{Q}_i^k, \psi_i^k >, \tag{8}$$

where $< \mathbb{Q}_i^k, \psi_i^k >$ is the locally trained model pair of client k in group $\mathcal{C}_i$, and $< \mathbb{Q}_i, \psi_i >$ is the global model pair of $\mathcal{C}_i$.

MRM creates an environment for *GSZMGs*. In this environment, we calculate the convergence bound of Oasis to theoretically demonstrate the effectiveness. It is worth noting that, for algorithms seeking NE, the proximity of NE is the convergence bound of the algorithm.

Theorem 4. *A GSZMG achieves a $(n\frac{\mathcal{O}(T)}{T})$-approximate NE, where n denotes the number of players, and T denotes iteration times.*

Proof. For an n-player *GSZMG* $\mathcal{G} =< V, E >$, let $A^{u,v}$ be the utility matrix for every player pair $(u,v) \in E$, z_u be the best response of player u, x_u be a mixed strategy for player u, and $\bar{x}_u^{(T)} = \frac{1}{T}\sum_{t=1}^{T} x_u^{(t)}$ for iteration times T. Our target is to prove that optimal payoff difference (*i.e.*, the difference between actual payoff and ideal payoff) $\Gamma := \sum_{(u,v) \in E} (z_u^T \cdot A^{u,v} \cdot \bar{x}_v^{(T)} - (\bar{x}_u^{(T)})^T \cdot A^{u,v} \cdot \bar{x}_v^{(T)})$ converge to some constant. Firstly, after clustering, we have $\Gamma \leq \sum_{i=0}^{k^*} \sum_{(v_i,u_i) \in E_i} (z_{u_i}^T \cdot A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)} - (\bar{x}_{u_i}^{(T)})^T \cdot A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)})$ for optimal number of clusters k^* representing the best quality-quantity tradeoff for the payoff of every player (*i.e.*, in a suboptimal quality-quantity environment, players are unable to obtain favorable payoff on their own). [18] draws a conclusion that within a group $\mathbb{C}_i$, $\sum_{(u_i,v_i) \in E_i} z_{u_i}^T \cdot$

$A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)} - \sum_{(u_i,v_i)\in E_i} (\bar{x}_{u_i}^{(T)})^T \cdot A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)} \leq |\mathbb{C}_i| \frac{\mathcal{O}(T)}{T}$.

We conclude that $\Gamma \leq \sum_{i=0}^{k^*} (\sum_{(v_i,u_i)\in E_i} z_{u_i}^T \cdot A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)} - \sum_{(v_i,u_i)\in E_i} (\bar{x}_{u_i}^{(T)})^T \cdot A^{u_i,v_i} \cdot \bar{x}_{v_i}^{(T)}) \leq \sum_{i=0}^{k^*} |\mathbb{C}_i| \frac{\mathcal{O}(T)}{T}$. Note that $\sum_{i=0}^{k^*} |\mathbb{C}_i| = n$, so $\Gamma \leq \sum_{i=0}^{k^*} |\mathbb{C}_i| \frac{\mathcal{O}(T)}{T} = n\frac{\mathcal{O}(T)}{T}$. Therefore, a *GSZMG* achieves a $(n\frac{\mathcal{O}(T)}{T})$-approximate NE. □

Through the proof of Theorem 4, we conclude that training federated GANs through Oasis can converge. The grouping environment created by **MRM** simplifies the game as players engage in *SZMGs* within the group, reducing unnecessary information effect and achieving global optima of all players under Oasis.

4.4 Local Regularization Module

Algorithm 2: *Loss-aware* JS-Regularized GANs

Input: Initial noise variance γ, annealing decay rate α, Δ *loss* threshold δ, number of local training iterations T, minibatch size m, number of steps to apply to the discriminator k

1 **for** $t = 1 : T$ **do**
2 **for** $s = 1 : k$ **do**
3 Sample minibatch of m examples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\} \sim \mathbb{P}$ from real data $p_{data}(\mathbf{x})$;
4 Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$ from noise prior $p_g(\mathbf{z})$;
5 Compute $F_\gamma(\mathbb{P}, \mathbb{Q}; \psi)$ according to Eq.10;
6 $\psi \leftarrow \psi + \nabla_\psi(F_\gamma(\mathbb{P}, \mathbb{Q}; \psi))$;
7 **end**
8 Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$ from noise prior $p_g(\mathbf{z})$;
9 Computing $F_\mathbb{Q}(\mathbb{Q}; \psi)$ according to Eq.12;
10 $\mathbb{Q} \leftarrow \mathbb{Q} - \nabla_\mathbb{Q} F_\mathbb{Q}(\mathbb{Q}; \psi)$;
11 **if** $F_\mathbb{Q}(\mathbb{Q}; \psi) < \delta$ **then**
12 $\gamma \leftarrow \gamma \cdot \alpha^{t/T}$; `/* loss-aware annealing */`
13 **end**
14 **end**

As discussed in Section 2.3.1, federated GANs match *PZPGs*, where it is highly computational to make every player get her NE. Therefore, we generalize *PZPGs* to *SZMGs*, and turn regularization as a coordinator for help to handle multiplayer interaction issue in federated GANs.

Local GAN training is equivalent to conducting *SZMGs* within groups, which requires that local generator and discriminator players do not pursue individual payoff maximization. Therefore, **LRM** needs to design a regularization term to modify the utility function (loss function) originally oriented towards maximizing individual player payoff.

An essence of the solution to linear programming for *SZMGs* $\mathcal{GG}$ is that players who choose strategies of adjacent nodes receive corresponding payoffs that the nodes would receive in $\mathcal{GG}$ from their joint interaction [18]. However, for GANs, determining the payoff for adjacent players is often challenging. For this issue, we provide a formal definition:

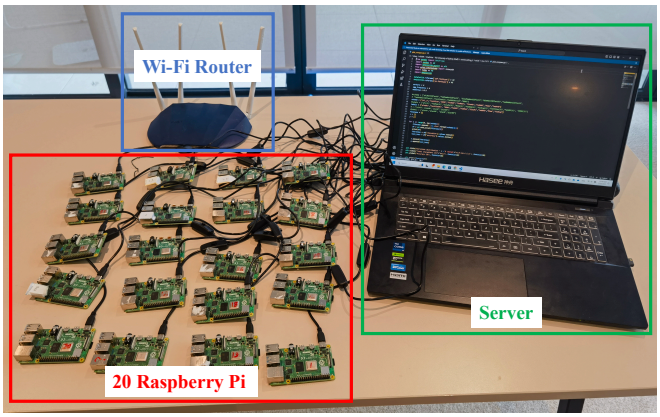


Fig. 10. Hardware prototype with the laptop being central server, 20 Raspberry Pi being devices. During the federated GAN experiments, the wireless router is placed 5 meters away from all the devices.

Problem 3. *(Non-supp) [35] and [36] show that* $supp(\mathbb{P}) \cap supp(\mathbb{Q})$ *is negligible whenever the model and/or data are confined to low-dimensional manifolds, which means* ***the discriminator and generator are unable to derive corresponding payoffs from their joint interaction****, resulting in the NE shift.*

To overcome *Non-supp*, we propose Algorithm 2. An intuitive idea is to train local GANs with white Gaussian noise Λ so that $\mathbb{P} * \Lambda$ as well as $\mathbb{Q} * \Lambda$ are sure to have full support in the ambient space:

$$F_\gamma(\mathbb{P}, \mathbb{Q}; \psi) := F(\mathbb{P} * \Lambda, \mathbb{Q} * \Lambda; \psi), \quad where\ \Lambda = \mathcal{N}(\mathbf{0}, \gamma\mathbf{I}). \tag{9}$$

[23] demonstrates that training GANs with noise is equivalent to regularization of the discriminator, which is achieved by integrating Jensen-Shannon (JS) Regularizer into the discriminator of the two-player oriented architecture. Therefore, we have the following regularization term for the discriminator:

$$F_\gamma(\mathbb{P}, \mathbb{Q}; \psi) = \mathbf{E}_\mathbb{P}[ln(\psi)] + \mathbb{Q}[ln(1-\psi)] - \frac{\gamma}{2}\Omega_{JS}(\mathbb{P}, \mathbb{Q}; \psi), \tag{10}$$

$$\Omega_{JS}(\mathbb{P}, \mathbb{Q}; \psi) = \mathbf{E}_\mathbb{P}[(1-\psi(x))^2 \|\nabla\phi(x)\|^2] + \mathbf{E}_\mathbb{Q}[\psi(x)^2 \|\nabla\phi(x)\|^2], \tag{11}$$

while the generator remains the same as proposed in the original work [1]:

$$F_\mathbb{Q}(\mathbb{Q}; \psi) = \mathbb{Q}[ln(1-\psi)]. \tag{12}$$

The process of local GAN training is the same as conventional central GAN training (line 2-10). Loss discrepancy *Δloss* threshold δ represents the expected overall payoff for the generator and discriminator, thereby mitigating the impact of regularization through annealing upon achieving δ (line 11-13). With JS Regularizer, players receive corresponding payoffs from their joint interaction, leading to improved attainment of global NE.

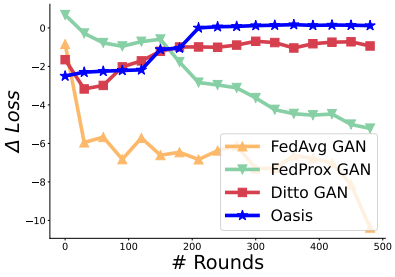


Fig. 11. Loss discrepancy comparison (MNIST dataset).

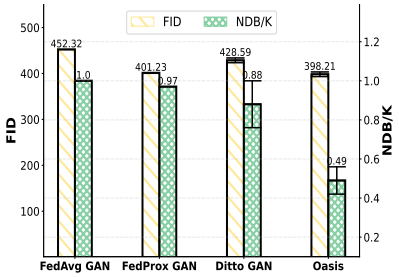


Fig. 12. Synthesis performance comparison (MNIST dataset).

Fig. 13. Random samples generated by Oasis (experiment results).

TABLE 2
Total time for 500 training rounds.

System	FedAvg GAN	FedProx GAN	Ditto GAN	Oasis
Time (h)	22.14	22.14	43.24	22.14

5 IMPLEMENTATION

We implement a prototype of Oasis based on an open-source federated learning platform *PFL-Non-IID* [37]. As a general federated learning platform, it typically targets the training of single models. However, for architectures like GANs, which consist of two sub-networks, we need to modify the `send_models` and `aggregate_parameters` functions to isolate the parameter distribution and aggregation for the generator and discriminator. Specifically, we overwrite these two functions to determine whether the current model is a generator or a discriminator based on the parameter type.

The Oasis server. For representations uploaded by the clients, the server first performs clustering. We utilize the `K-means` function and `silhouette_score` function implemented by scikit-learn [38]. For the `K-means` function, we use the Euclidean distance as the similarity measure, while for the `silhouette_score` function, we employ the correlation coefficient as the cluster distance metric. After generating the similarity table, the training of federated GANs begins in Oasis. We modify `aggregate_parameters` function and `send_models` function. For the `aggregate_parameters` function, Oasis examines the group IDs of the clients and averages the local model pairs of clients with the same group ID. As for the `send_models` function, after randomly selecting clients for training, Oasis checks their group IDs and sends the corresponding group's global model to each selected client.

The client. Before starting the training process, client data representations are firstly extracted. For feature extraction, we randomly select K batches of data and average them to obtain a point in a high-dimensional space. In the case of multi-channel data, we average values across multiple channels to obtain a unified representation. Then Oasis initiates a thread to upload the data representations to the server. During local GAN training, the `model_trainer` function provides training loss to **LRM**, and in turn, the `model_trainer` function receives regularization terms from **LRM** to assist with training.

6 EXPERIMENT

In this section, we validate the effectiveness of the proposed system through experiments. We initially conduct experiments on Raspberry Pi to verify the performance in real-world scenarios. We start by presenting the evaluation setup, and then show the experimental results.

6.1 Experimental Setup

Platform. We initially performed experiments on a networked hardware prototype system. The prototype, depicted in Fig. 10, comprises 20 Raspberry Pis (version 4) with 4GB of RAM functioning as devices and a laptop computer acting as the central server. The devices are interconnected through an enterprise Wi-Fi router, and we establish a TCP-based socket interface for peer-to-peer connections. We expand the RAM of the Raspberry Pi to 8GB by enabling virtual memory. Throughout the process, we utilize the 5G frequency band for parameter transmission.

FL Dataset and Model. Training GANs is resource-intensive, and Raspberry Pi has limited hardware resources. Therefore, in real-world scenarios, we use the MNIST [39] dataset due to its compatibility with the hardware constraints. For the GAN model, we opt for MLPGAN [1].

Compared Methods. The baselines we selected include *FedAvg*, *FedProx* aimed at global optima, and *Ditto* based on the grouping concept.

1) *FedAvg*. FedAvg is the original decentralized learning algorithm used in FL. We act FedAvg as a baseline to compare the performance of different FL methods.
2) *FedProx* [9]. FedProx can be seen as a generalization and re-parametrization of FedAvg by introducing a proximal term to the objective function to achieve global optima.
3) *Ditto* [10]. Ditto is a multi-task learning objective framework for FL that provides *personalization* based on grouping concepts while retaining similar efficiency and privacy benefits as FL.

We integrate these three methods into the federated GAN system. For FedProx, we set proximal term coefficient μ to 0.5. For Ditto, we set μ to 0.1 and the personalized local step to 5. For Oasis, we set η to 2000, γ to 0.1, and for further discussion on Oasis hyperparameters, please refer to Section 7.4, where we conduct a series of ablation evaluation to demonstrate the effect of each module.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

TABLE 3
FID comparison on various datasets.

Dataset / Algorithm	*IN*	MNIST	Cifar10	CelebA
FedAvg	5k	529.32	144.12	147.90
	10k	529.32	142.70	147.05
FedProx	5k	463.37	325.02	**109.60**
	10k	463.37	323.79	**108.69**
Ditto	5k	469.66±*45.02*	186.12±*47.89*	429.65±*0.46*
	10k	469.63±*45.04*	184.77±*48.10*	428.97±*0.40*
Ours	5k	**443.70±*27.92***	**119.92±*15.64***	128.21±*9.07*
	10k	**443.27±*28.12***	**118.88±*15.49***	127.32±*9.20*

TABLE 4
NDB/K comparison on various datasets.

Dataset / Algorithm	*K*	MNIST	Cifar10	CelebA
FedAvg	100	1.00	0.82	0.10
	200	1.00	0.68	0.08
FedProx	100	1.00	0.99	**0.02**
	200	1.00	1.00	**0.04**
Ditto	100	0.99±*0.01*	0.84±*0.10*	0.10±*0.01*
	200	0.99±*0.02*	0.79±*0.13*	0.06±*0.01*
Ours	100	**0.49±*0.24***	**0.78±*0.07***	0.04±*0.01*
	200	**0.38±*0.27***	**0.66±*0.03***	0.05±*0.01*

Metrics. We employ two metrics to assess the synthesis performance of GANs. The first metric is FID [20], which quantifies the overall semantic realism of synthesized data. FID serves as a measure of the quality of generated data. The second metric is NDB/K [21], which indicates the number of distinct categories or intervals that the data can be divided into based on statistically significant differences. NDB/K provides an indication of the variety of generated data.

6.2 Performance Results

Overall Performance. Fig. 11 illustrates NE shift for four federated GAN systems. As discussed in Section 2.1, we measure NE shift using the $\Delta Loss$, where a larger one indicates a closer proximity to the optimal NE. It is observed that, after several rounds of game play, our proposed Oasis is closest to the optimal NE, followed by Ditto GAN. In contrast, FedAvg GAN and FedProx GAN fail to converge to the NE. In Oasis, before 200 rounds, some clients' data has not been sampled, leading to a decrease in $\Delta Loss$. After 200 rounds, all clients have been sampled, causing the $\Delta Loss$ of Oasis to begin rising and approach zero. This trend can also be observed in Ditto, another personalized model approach, where $\Delta Loss$ first decreases and then increases. We believe that in Oasis, by 200 rounds, all clients have been sampled, thus making the effectiveness of Oasis's personalized model method more apparent. Fig. 12 illustrates the synthesis performance of the generators trained by four federated GAN systems. It is observed that Oasis achieves the best performance in both data quantity and variety, while the generators trained by the other three systems fail to capture the true data distribution. Compared to FedAvg GAN, both FedProx GAN and Ditto GAN have improved data quality (lower FID) and data diversity (lower NDB/K) to some extent. However, in terms of FID, Oasis is 0.75% lower than FedProx GAN and 7.13% lower than Ditto GAN. For NDB/K, Oasis is 49.48% lower than FedProx GAN and 44.32% lower than Ditto GAN. Fig. 13 illustrates the MNIST images generated by Oasis. The generated images clearly distinguish between different digits, indicating high data quality. Additionally, all digits except for 5 are generated, demonstrating good data diversity. This further validates the effectiveness of our proposed system.

Training Time Analysis. In Table 2, we record the total time taken to train GANs for 500 rounds on our hardware platform using four different systems. The Raspberry Pi training time per step is approximately 30.38 seconds, and the communication time between the server and clients is about 3.75 seconds. Both FedAvg GAN and FedProx GAN do not have additional components and execution rounds, so their training times are consistent. Ditto GAN has the longest total training time due to additional local training steps. The time taken by Oasis to extract data representations is about 0.22 seconds on clients, the time to upload these data representations is about 0.25 seconds, and the time to cluster the data representations on the server to obtain the similarity table is about 3.86 seconds. Although Oasis requires extracting data representations and uploading them (see Section 7.5 for a detailed analysis) and involves grouping aggregation operations on the server, when compared to client training time, these are negligible. Therefore, Oasis improves training performance without increasing training duration.

7 Simulation

In this section, we validate the effectiveness of the proposed system through simulation. We scale up the dataset and client quantity to simulate the system's behavior in a high-load environment. We start by presenting the simulation setup, and then show the simulation results.

7.1 Simulation Setup

Platform. We prototype Oasis and conduct the simulation evaluation on a server equipped with four NVIDIA A100 Tensor Core (40G) GPU cards, an Intel(R) Xeon(R) Gold 6240C CPU (@ 2.60GHz), and 256GB of memory.

FL Datasets. We create data heterogeneity via the dirichlet distribution [40], which ensures each client possesses a small portion of a specific type of data while also holding a substantial amount of data from other categories. Specifically, we employ three datasets, *i.e.*, MNIST, Cifar10 [41], and CelebA [42], which consist of images with varying sizes and different numbers of channels. By exploiting them with various GAN implementations, we conduct comprehensive simulation to evaluate effectiveness of our designed system.

Models. We use the following GAN models: MLPGAN for MNIST, DCGAN [43] for Cifar10, and U-NetGAN [44] for CelebA. MLPGAN is composed of fully connected layers, leaky ReLU activations, and tanh output. The architecture of DCGAN includes convolutional and transposed convolutional layers, along with batch normalization and

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148

Fig. 14. Random samples generated by Oasis trained on MNIST, Cifar10, and CelebA (simulation results).

activation functions. U-NetGAN leverages the skip connections and symmetrical structure of the U-Net architecture to enable more effective discrimination, leading to improved performance and the generation of higher-quality samples.

Client Quantity. For the MINST and Cifar10 datasets, we conduct simulation with 100 clients. However, for the CelebA dataset, given its large data size and complex features, we perform simulation with 10 clients. It is noteworthy that on a Raspberry Pi with 4GB of RAM, we encounter limitations preventing the execution of U-NetGAN for CelebA.

The **comparison methods** and **metrics** are consistent with what is described in Section 6.

7.2 Overall Performance

We investigate data generation ability of trained GANs to evaluate performance of the four FL approaches. In Table 3 and Table 4, we present the FID scores and NDB/K scores of the trained generators after using different FL approaches and datasets to train GANs. We calculate FID using the whole training set and two different generated Image Numbers (*IN*). For NDB/K calculation, we consider two values of K. In both tables, as Ditto and Oasis train multiple generators, we represent the results using mean $\pm$ *std*. In the MNIST and Cifar10 datasets, Oasis achieves the best performance overall. Compared to FedAvg, it demonstrates an average improvement of 16.48% on FID and 30.20% on NDB/K, which can be attributed to the effective use of *Regularization* and *Clustering* techniques employed in Oasis. While FedProx exhibits the poorest performance on the two datasets, it achieves the best performance on the CelebA dataset, which is ascribed to the fact that the CelebA dataset lacks the concept of labels, thus failing to constitute a data heterogeneous environment with imbalanced label distributions, and making Oasis **FSC** module not effectively function. However, Oasis still outperforms the other two approaches by an average improvement of 41.80% on FID and 43.54% on NDB/K, which is attributed to the fact that like FedProx, Oasis also incorporates a regularization module. We observe that Ditto does not fully resolve vanishing gradient and mode collapse, which indicates that training GANs using Ditto can lead to local optima, failing to capture the entire data distribution and achieve global optima. In contrast, our proposed system, Oasis, achieves an average overall improvement of 23.13% on FID and 26.33% on NDB/K, demonstrating its ability to simultaneously address vanishing gradient and mode collapse. Fig. 14 showcases the MNIST, Cifar10, and CelebA images generated by Oasis, which appear authentic and diverse.

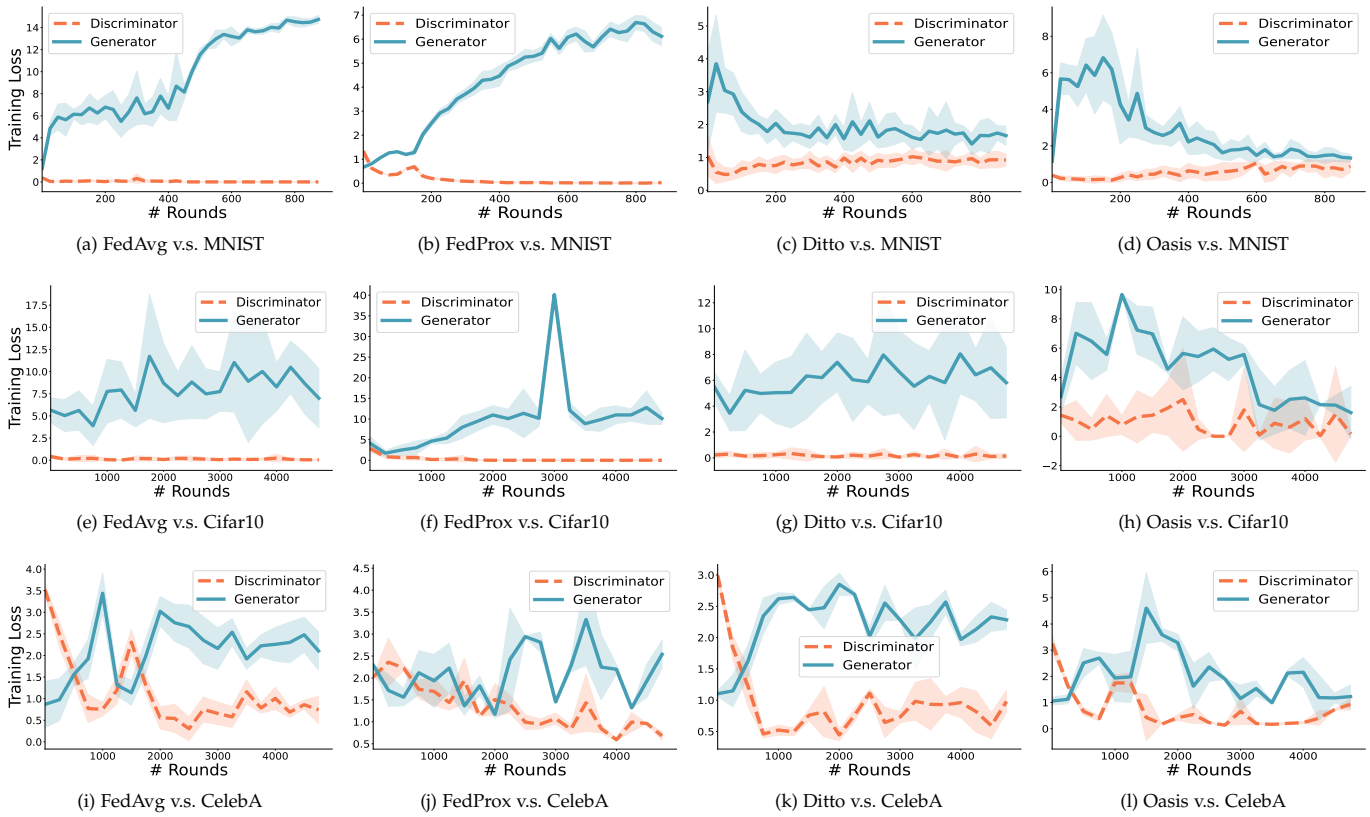
7.3 Analysis of Training Convergence

In Fig. 15, we present the training loss of MLPGAN for MNIST, DCGAN for Cifar10, and U-NetGAN for CelebA on FedAvg, FedProx, Ditto, and Oasis in a non-*i.i.d.* setting. Fig. 15(a) shows that loss discrepancy of the generator and discriminator in FedAvg is around 16, indicating severe NE shift, *i.e.*, vanishing gradient. FedProx reduces the loss discrepancy by an average of 18.75% but increases training instability (*e.g.*, Fig. 15(f) and 15(j)), still failing to resolve the NE shift. The training outcomes of FedAvg and FedProx demonstrate that in a non-*i.i.d.* environment, training a single global GAN model cannot capture the data distribution of all clients, highlighting the necessity of grouping or personalization. Ditto (see Fig. 15(c), 15(g) and 15(k)), as a personalized FL approach, and Oasis, which leverages *Clustering*, are investigated. Although Oasis exhibits a gradual increase in loss discrepancy during the early rounds, it progressively reduces and ultimately approaches the NE on average 42.27% more closely than Ditto. Fig. 15(d), 15(h) and 15(l) show that Oasis succeeds in training GANs that converges and approaches NE, demonstrating promising training efficacy.

7.4 Ablation Evaluation

We conduct a series of ablation evaluations and discuss the impact of Oasis hyperparameter selection on the evaluation metrics (FID, NDB/K, and Loss Discrepancy) in Table 5. Without loss of generality, we conduct simulation using the MNIST dataset. We modify cluster number k by setting different quantity thresholds η. Additionally, we investigate the influence of the regularization term by varying the noise variance γ. We choose three quantity thresholds: 6000, 200, and 50, resulting in cluster numbers of 1, 7, and 100, respectively, which are abbreviated as $\eta \rightarrow k$. Among these, the cluster number 7 achieves the highest silhouette score, and is thus the optimal cluster number. Cluster number 1 represents all clients in a single cluster, while cluster number 100 assigns each client to a separate cluster. We set two noise variances, 0.0 and 0.1, to explore the impact of the regularization term. Intuitively, with a number of clusters of 1 and a regularization term of 0, Oasis degenerates into FedAvg. We observe that both excessively large and small cluster quantities affect training performance, as mentioned in Section 4.2, emphasizing the importance of balancing data quality and quantity. The addition of the regularization term benefits the training process, leading to an average improvement of 6.30% on FID, 27.31% on NDB/K, and 58.40% on loss discrepancy compared to scenarios without the regularization term. The best training results are achieved when the cluster number is 7 (see Fig. 16) and the noise variance is 0.1, with an FID of 433.13$\pm$*6.32*, an NDB/K of 0.38$\pm$*0.18*, and a loss discrepancy of -0.51$\pm$*0.21*.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3438148



(a) FedAvg v.s. MNIST (b) FedProx v.s. MNIST (c) Ditto v.s. MNIST (d) Oasis v.s. MNIST

(e) FedAvg v.s. Cifar10 (f) FedProx v.s. Cifar10 (g) Ditto v.s. Cifar10 (h) Oasis v.s. Cifar10

(i) FedAvg v.s. CelebA (j) FedProx v.s. CelebA (k) Ditto v.s. CelebA (l) Oasis v.s. CelebA

Fig. 15. Training loss of three specific GAN implementations on FedAvg, FedProx, Ditto, and Oasis over three different datasets (non-*i.i.d.* setting).

TABLE 5
Impact of hyperparameter selection.

$\eta \rightarrow k, \gamma$	FID↓	NDB/K↓	$\Delta Loss$↑
$6000 \rightarrow 1, 0.0$	526.61	1.00	-14.43
$6000 \rightarrow 1, 0.1$	440.16	0.63	-2.77
$200 \rightarrow 7^*, 0.0$	441.85±*14.74*	0.69±*0.21*	-3.87±*0.55*
$200 \rightarrow 7^*, 0.1$	**433.13±*6.32***	**0.38±*0.18***	**-0.51±*0.21***
$50 \rightarrow 100, 0.0$	457.50±*14.18*	0.97±*0.03*	-4.62±*1.60*
$50 \rightarrow 100, 0.1$	455.10±*15.05*	0.97±*0.03*	-4.27±*1.09*

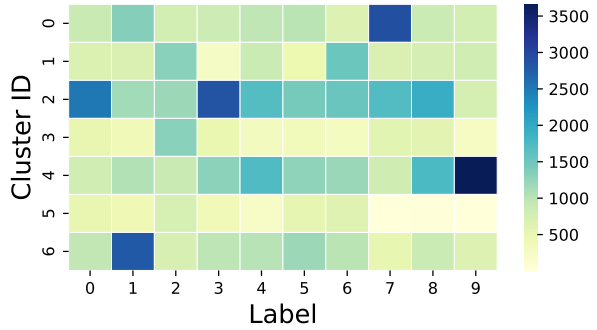


Fig. 16. An illustration of $k^* = 7$ with rectangles representing the count of data samples per label within clusters.

7.5 Exploration of Extra Communication Overhead

Our approach involves uploading data representations to the server, which results in additional communication costs. Table 6 presents a detailed breakdown of the parameter quantities required for the upload. It displays the parameter quantities (**#Params** row) of three GAN models along with their corresponding data representation (dr) parameter (**Item** row) quantities. The size of representation parameters, on average, accounts for only 24.91% of the overall model parameters' size, which means that the representation information is relatively insignificant compared to the model itself. This suggests that the overall increase in communication costs would be minimal.

TABLE 6
The number of parameters for uploading.

Item	MLPGAN	dr	DCGAN	dr	U-NetGAN	dr
#Params	11.24M	0.83M	4.06M	0.09M	258.77M	168.56M

8 Related Work

Clustered Federated Learning. Data heterogeneity poses a challenge in FL. Clustering is a method to address this issue. The fundamental idea of clustered FL is to partition tasks into clusters, where tasks within each cluster share more similarity in terms of model parameters. Clustered FL can be categorized into one-shot clustering (*e.g.*, [29], [30]) and iterative clustering (*e.g.*, [27], [28]). One-shot clustering separates the task clustering and parameter learning into two stages, while iterative clustering simultaneously learns task clustering and model parameters during training iterations. The advantage of one-shot clustering lies in its low communication overhead and the ability to consider clustering algorithms with higher time complexity for clustering is performed only once. However, its drawback is the inability

to adapt and adjust to changes in clients during training. On the other hand, the advantage of iterative clustering is that the model can adapt and improve its accuracy based on changing conditions. However, it requires longer convergence time and additional clustering metrics during communication. The metrics of clustering include gradients (training loss) and indirect data patterns. As discussed in Section 4.1, federated GANs are not suitable for the former. In this work, we employ a one-shot clustering approach, which is specifically designed for generative models. Note that, for data representation-based clustered FL, a one-shot clustering is sufficient, as the data remains unchanged during the training process. We exploit STL and indirectly extract data representations with privacy preservation as clustering metrics. Therefore, unlike most one-shot clustering methods that utilize loss or parameter gradient similarity as clustering metrics, our approach minimizes communication overhead and is well-suited for generative models.

Multiplayer Oriented Generative Adversarial Networks. GANs corresponds to two-player zero-sum minimax games. [18] has revealed that finding NE in $three$-player zero-sum games is already PPAD-complete, implying that achieving NE and convergence in multiplayer oriented GANs is challenging and time-consuming. To address this issue, [15], [16] have proposed novel aggregation approaches. However, these approaches primarily focus on statistics, whereas our approach is grounded in game theory and addresses the problem theoretically and at its root. [45] demonstrates the existence of a duality gap in two-team zero-sum games compared to two-player zero-sum games. Building upon this insight, [25] formulates multi-agent GANs as two-team games and introduces a first-order method that integrates control theory techniques to converge to NE. Nevertheless, we emphasize that this formulation is unsuitable for applying GANs to FL paradigms. This is due to the fact that in FL paradigms, players only communicate with each other through parameters/gradients fusion on the server, which does not establish a comprehensive two-team environment. In this work, we create a communication environment for multiplayer oriented GANs by grouping players together. Moreover, within groups, we leverage regularization methods to encourage players to reduce individual payoff maximization and share more information, achieving global optima and overall payoff maximization.

9 Conclusion

In this paper, we study training federated GANs. We present a comprehensive motivation, highlighting the NE shift in multiplayer oriented GANs, exacerbated by data heterogeneity, leading to poor training performance with issues of vanishing gradient and mode collapse. We propose the Oasis system, which employs *Regularization* and *Clustering* techniques to mitigate the training performance issues. We formulate the training of federated GANs as *Group-wise Separable Zero-sum Multiplayer Games* and demonstrate that the game achieves a $(n\frac{\mathcal{O}(T)}{T})$-approximate NE. We compare Oasis against three other *state-of-the-art* FL approaches both on a hardware prototype and in a simulated environment, showcasing improvements in convergence and synthesis performance.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[2] X.-Y. Liu and X. Wang, "Real-time indoor localization for smartphones using tensor-generative adversarial nets," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3433–3443, 2020.

[3] Q. Cui, Z. Zhou, Z. Fu, R. Meng, X. Sun, and Q. J. Wu, "Image steganography based on foreground object generation by generative adversarial networks in mobile edge computing with internet of things," *IEEE Access*, vol. 7, pp. 90 815–90 824, 2019.

[4] M. Amin, B. Shah, A. Sharif, T. Ali, K.-I. Kim, and S. Anwar, "Android malware detection through generative adversarial networks," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 2, p. e3675, 2022.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[6] D. Zhang and A. Khoreva, "Pa-gan: Improving gan training by progressive augmentation," 2018.

[7] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Federated anomaly analytics for local model poisoning attack," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 596–610, 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[10] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[11] Z. Wang, H. Ji, Y. Zhu, D. Wang, and Z. Han, "A survey on federated analytics: Taxonomy, enabling techniques, applications and open issues," *arXiv preprint arXiv:2404.12666*, 2024.

[12] D. Ravichandran and S. Vassilvitskii, "Evaluation of cohort algorithms for the floc api," *Google Research & Ads white paper, available at: http://githubusercontent. com/google/ads-privacy/master/proposals/FLoC/FLOC-Whitepaper-Google. pdf (accessed 5th February, 2021)*, 2021.

[13] Z. Wang, Y. Zhu, D. Wang, and Z. Han, "Fedfpm: A unified federated analytics framework for collaborative frequent pattern mining," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 61–70.

[14] Q. Pan and Y. Zhu, "Fedwalk: Communication efficient federated unsupervised node embedding with differential privacy," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1317–1326.

[15] W. Li, J. Chen, Z. Wang, Z. Shen, C. Ma, and X. Cui, "Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] R. Guerraoui, A. Guirguis, A.-M. Kermarrec, and E. L. Merrer, "Fegan: Scaling distributed gans," in *Proceedings of the 21st International Middleware Conference*, 2020, pp. 193–206.

[17] D. Saxena and J. Cao, "Generative adversarial networks (gans) challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[18] Y. Cai and C. Daskalakis, "On minmax theorems for multiplayer games," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2011, pp. 217–234.

[19] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[21] E. Richardson and Y. Weiss, "On gans and gmms," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[22] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," *Advances in neural information processing systems*, vol. 29, 2016.

[23] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," *Advances in neural information processing systems*, vol. 30, 2017.

[24] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[25] F. Kalogiannis, I. Panageas, and E.-V. Vlatakis-Gkaragkounis, "Towards convergence to nash equilibria in two-team zero-sum games," in *The Eleventh International Conference on Learning Representations*, 2022.

[26] L. Bergman and I. Fokin, "On separable non-cooperative zero-sum games," *Optimization*, vol. 44, no. 1, pp. 69–84, 1998.

[27] U. Zurutuza, "Clustered federated learning architecture for network anomaly detection in large scale heterogeneous iot networks," 2023.

[28] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 54–66.

[29] Y. Cao, S. Maghsudi, T. Ohtsuki, and T. Q. Quek, "Mobility-aware routing and caching in small cell networks using federated learning," *IEEE Transactions on Communications*, 2023.

[30] J. Liu, J. Yan, H. Xu, Z. Wang, J. Huang, and Y. Xu, "Finch: Enhancing federated learning with hierarchical neural architecture search," *IEEE Transactions on Mobile Computing*, 2023.

[31] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.

[32] J. Torres-Sospedra, P. Richter, A. Moreira, G. M. Mendoza-Silva, E. S. Lohan, S. Trilles, M. Matey-Sanz, and J. Huerta, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 769–782, 2020.

[33] H. Chen, I. Chillotti, Y. Dong, O. Poburinnaya, I. Razenshteyn, and M. S. Riazi, "{SANNS}: Scaling up secure approximate {k-Nearest} neighbors search," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2111–2128.

[34] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[35] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[36] H. Narayanan and S. Mitter, "Sample complexity of testing the manifold hypothesis," *Advances in neural information processing systems*, vol. 23, 2010.

[37] TsingZ0, "Pfl-non-iid," 2023. [Online]. Available: https://github.com/TsingZ0/PFL-Non-IID

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] Y. LeCun, "The mnist database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[40] S. Shi, Y. Guo, D. Wang, Y. Zhu, and Z. Han, "Distributionally robust federated learning for network traffic classification with noisy labels," *IEEE Transactions on Mobile Computing*, 2023.

[41] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[42] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.

[43] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[44] E. Schonfeld, B. Schiele, and A. Khoreva, "A u-net based discriminator for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8207–8216.

[45] L. J. Schulman and U. V. Vazirani, "The duality gap for two-team zero-sum games," *Games and Economic Behavior*, vol. 115, pp. 336–345, 2019.

Chuang Hu received his BS and MS degrees from Wuhan University in 2013 and 2016. He received his Ph.D. degree from the Hong Kong Polytechnic University in 2019. He is currently an Associate Professor in the School of Computer Science at Wuhan University. His research interests include edge learning, federated learning/analytics, and distributed computing.

Tianyu Tu received the B.S. degree in Software Engineering from Wuhan University in 2022. He is currently working towards the M.S. degree in Computer Science and Technology with the Computer School, Wuhan University. His research interests include federated unsupervised learning and game theory.

Yili Gong received her BS degree in Computer Science from Wuhan University in 1998, and her Ph.D. degree in Computer Architecture from the Institute of Computing, Chinese Academy of Sciences in 2006. She is currently an Associate Professor in the School of Computer Science at Wuhan University. Her interests include intelligent operations and maintenance in HPC environments, distributed file systems and blockchain systems.

Jiawei Jiang received his PhD degree in computer science from Peking University, China, in 2018. He is currently a professor with the School of Computer Science, Wuhan University, China. His research interests include database, big data management and analytics, and machine learning systems.

Zhigao Zheng received his PhD degree in computer science from Huazhong University of Science and Technology. He is currently with the School of Computer Science, Wuhan University, China. His research interests focus on cloud computing, big data processing, and AI systems. He is the editorial board member of some high-quality journals, such as the IEEE Transactions on Consumer Electronics, Mobile Networks and Applications.

Dazhao Cheng (Senior Member, IEEE) received his BS and MS degrees in Electrical Engineering from the Hefei University of Technology in 2006 and the University of Science and Technology of China in 2009. He received his Ph.D. from the University of Colorado at Colorado Springs in 2016. He was an AP at the University of North Carolina at Charlotte in 2016-2020. He is currently a professor in the School of Computer Science at Wuhan University. His research interests include big data and cloud computing.