Decentralized Data Synthesis from Private Non-IID Data[*]

Muhammad Usama Saleem
Department of Computer Science
UNC Charlotte
Charlotte, USA
msaleem2@uncc.edu

Liyue Fan
Department of Computer Science
UNC Charlotte
Charlotte, USA
liyue.fan@uncc.edu

Abstract—**Privacy-preserving data sharing enables a wide range of exploratory and secondary data analyses while protecting the privacy of individuals in the datasets. Recent advancements in machine learning, specifically generative adversarial networks (GANs), have shown great promise for synthesizing realistic datasets. In this work, we investigate the feasibility of training GAN models privately in practical settings, where the input data is distributed across multiple parties, and local data may be highly skewed, i.e., non-IID. We examine centralized private GAN solutions applied at each local party and propose a federated solution that provides strong privacy and is suitable for non-IID data. We conduct extensive empirical analysis with a wide range of non-IID settings and data from different domains. We provide in-depth discussions about the utility of the synthetic data, the privacy risks in terms of membership inference attacks, as well as the privacy-utility trade-off for private solutions.**

Index Terms—**Federated Learning, Differential Privacy, Non-IID Data, Generative Adversarial Networks**

I. Introduction

Sharing individual-level data is beneficial for collaborative analyses and advancing research. However, sharing individually contributed data often gives rise to privacy concerns, as even “anonymized” data can be re-identified by linking external databases [1], or by examining unique behaviors, e.g., [2] and [3]. Data synthesis, which generates *fake* records that capture the characteristics of real data, provides great promise for privacy-protecting data sharing. Among existing data synthesis techniques, Generative Adversarial Networks (GANs) [4] have become the state-of-the-art approach for learning generative models and providing high-quality synthetic data. They have been successfully applied to synthesizing a variety of data types, such as tabular and imaging data.

To deploy GANs for privacy-protecting data synthesis, there are several practical challenges. Firstly, GAN models do not provide guarantees on what the models and generated data may reveal about real, sensitive training data. Recent research has shown successful membership inference attacks on GAN models [5], [6], where the participation of target individuals may be inferred. Furthermore, real training data for GAN models are often distributed across multiple parties, e.g., hospitals and banks, which may not be shared with external parties. Moreover, local data often comes from non-identical distributions, which may pose significant challenges in machine learning applications [7].

To address those challenges, we propose two approaches for private synthesis from decentralized non-IID data using GANs. Furthermore, we adopt differential privacy to provide rigorous privacy guarantees for input training data. As shown in Figure 1, the first approach allows each party to train local GAN models privately and only share synthetic samples. The second approach leverages recent development in differential privacy and federated learning and learns private GAN models collaboratively with all parties. To address non-IID local data distributions, we build our federated approach on a recently proposed framework [8], which improves training convergence by taking into account statistical heterogeneity among local parties. To understand the practical privacy protection offered by our proposed solutions, we adopt a range of membership inference attacks and assess their accuracy. We conduct an extensive empirical evaluation with data from multiple image domains and in a variety of simulated non-IID settings. Our results help readers understand the utility and privacy properties of the proposed solutions, as well as the effects of practical factors, such as the privacy requirement, the number of parties, and the level of skewness.

The rest of the paper is organized as follows: Section II briefly describes recent work on generative models, privacy in data synthesis, and federated learning frameworks; Section III introduces the technical definition of differential privacy and generative adversarial networks; Section IV presents our proposed approaches to private data synthesis and non-IID data simulation strategies; Section V describes practical utility and privacy measures adopted in the problem setting; Section VI presents a comprehensive empirical evaluation with real-world datasets; Section VII provides interpretation of the results and points out considerations for deployment; Section VIII concludes the paper and states future research directions.

II. Related work

Generative adversarial networks (GANs) [4] have demonstrated significant potential for learning complex distributions and generating realistic datasets in a wide range of domains.

[*] This research has been supported in part by National Science Foundation grants CNS-1951430, CNS-2027114, and CNS-2144684. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

GANs learn a generator and a discriminator in an adversarial process to produce high-quality synthetic data. Recent research addresses challenges in learning GAN models, such as mode collapse, instability, and architectural design [9].

Due to memorization and over-fitting, machine learning models are prone to a range of privacy risks, such as membership inference and model inversion. By accessing the trained models, attackers may infer the participation of a target individual [10] and even reconstruct representative training data [11]. Several recent studies have synthesized various attacks on membership inference for general machine learning models [12] and GAN models specifically [6]. While cryptographic approaches, such as [13], [14], have been proposed to enable machine learning over private data, they incur high computation and/or communication overheads. Furthermore, privacy attacks may be launched when the learned models are shared with untrusted recipients. As suggested in [10], differential privacy [15], [16] is a promising solution to safeguard training data in machine learning applications. Several approaches, surveyed in [17], have been proposed to learn GAN models with differential privacy in centralized settings.

Federated learning [18] has emerged as a popular solution for training machine learning models when input data is distributed across multiple parties. Without sharing local training data, each party computes a local update to the global model and shares the update with a central server; the server keeps the global model up-to-date by aggregating all local updates. However, when local parties have highly skewed, non-IID data, the standard federated algorithm FedAvg [18] has shown to perform poorly empirically [19]. To address statistical heterogeneity among parties, FedProx [8] was proposed to improve the convergence on learning over data from non-identical distributions. There have been efforts to provide differential privacy guarantees for participants of federated learning [20]. In this work, we investigate the effects of differential privacy on federated data synthesis with non-IID distributions, which have not been well understood.

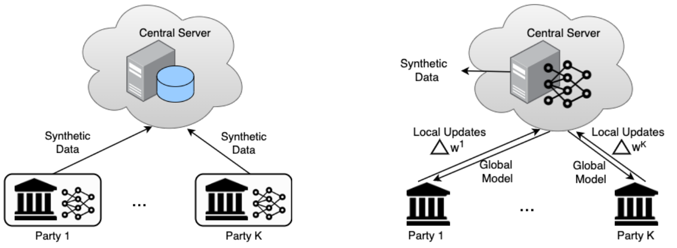
III. Background

In this section, we will introduce differential privacy and generative adversarial networks.

Differential Privacy. Differential privacy (DP) [15] is the state-of-the-art notion for providing privacy protection in statistical databases. It allows the release of aggregate statistics about the input database using randomized mechanisms, such that the output is roughly the same by adding or removing an individual record. Formally, given two neighboring databases $\mathcal{D}$ and $\mathcal{D}'$ which differ by at most one record, a randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy [15] if for any $\mathcal{Z} \subset range(\mathcal{M})$,

$$\Pr[\mathcal{M}(D) \in Z] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in Z] + \delta \tag{1}$$

The parameters $\epsilon > 0$ and $\delta \in [0,1]$ specify the level of privacy. Often referred to as the privacy *budget*, smaller ϵ and δ values indicate stronger privacy, and vice versa.



(a) Local Training (b) Federated Training

Fig. 1: Approaches to Private Data Synthesis from Decentralized Non-IID Data: note that in both approaches, the central server may be potentially untrusted.

One property of differential privacy is its *composability*. To keep track of the privacy budget spent in machine learning, a privacy accountant based on moment accounting [16] has been proposed. To give tighter bounds for privacy loss, the Rényi Differential Privacy (RDP) Accountant [21] and a subsampled RDP method [22] were proposed recently. In this work, we adopt the RDP accountant for more accurate privacy estimation and report the resulting ϵ and δ values.

While classic DP protects individual records (e.g., [16]), there is a need for providing *user-level privacy* [15] when each user contributes more than one record to the dataset, e.g., multiple user images for training a face recognition classifier. Recent work [20] implements user-level privacy in federated learning via local clipping and global perturbation, such that the participation of a user is protected. We note that user-level privacy is stronger than record-level privacy and may lead to higher utility loss.

GANs. Generative Adversarial Networks (GANs) [4] are state-of-the-art approaches for learning generative models and producing realistic synthetic data. GAN models include a generator G and a discriminator D. The generator's objective is to produce realistic samples that can fool the discriminator, while the discriminator's goal is to distinguish between generated samples and real ones. Let p_z be G's input noise distribution and p_{data} be the real data distribution. The generator G and discriminator D are trained in a two-player minimax game with the following objective:

$$\min_{\{G\}} \max_{\{D\}} \underbrace{E_{x\sim p_{data}}[\log(D(x))] + \overbrace{E_{z\sim p_z}[\log(1 - D(G(z)))]}^{\text{generator loss } l_G}}_{\text{discriminator loss } l_D} \tag{2}$$

Variants of the original GAN formulation have been proposed to improve training and incorporate auxiliary information [23], [24]. Differential privacy has been applied in learning GAN models in centralized settings [17], i.e., where all training data is located at one party while providing rigorous privacy guarantees.

IV. Proposed Approaches

In this work, we address privacy-preserving data synthesis via GANs in practical settings, where training data D is distributed across K ($K > 1$) parties, i.e., $D = \bigcup_{k=1}^{D} D^k$, and local data distributions may be skewed, i.e., non-IID. We assume that local datasets do not overlap, i.e., $D^i \cap D^j = \emptyset$ for $i \neq j$. Specifically, we consider two different approaches as shown in Fig. 1: (1) local GAN models are trained privately by each party; (2) global GAN models are trained privately in a federated framework.

A. Training GANs Locally

The first approach allows each party to train local GAN models using private solutions designed for the centralized setting. As a result, privacy protection is provided to local records, independent of external factors. After training, synthetic records sampled from private local generators can be shared with a central server, which forms a repository of synthetic data.

Specifically, we consider two existing solutions for this approach, i.e., DPGAN [25] and privGAN [26]. DPGAN [25] applies the DP-SGD method [16] in training GAN models, which provides differential privacy guarantees to real samples. DPGAN has been applied to synthesize a range of data types in centralized settings [17]. However, due to the perturbation introduced by differential privacy, the synthetic samples often show lower quality compared to those of non-private GAN models. Recently, privGAN [26] was proposed to defend against membership inference attacks specifically. Its main idea is to reduce the memorization of training samples by training separate GAN models on disjoint partitions along with a privacy discriminator to infer the generator of a given synthetic sample. The overall objective is a weighted sum of those of GAN models and the privacy discriminator, with a multiplicative factor λ for the privacy discriminator. Lower λ leads to reduced memorization and often lower quality. While privGAN does not provide any privacy guarantees, it reported good utility and efficacy against membership inference [26].

We propose to train DPGAN or privGAN models at each local party with pre-defined privacy parameters, e.g., ϵ for DPGAN and λ for privGAN, in order to provide equivalent privacy protection across parties. The evaluation measures for utility and privacy will be introduced in the following section.

B. Federated GAN Training

Federated learning has emerged as a promising solution for jointly training machine learning models while the input data is distributed across multiple parties. One advantage of the federated learning paradigm is that it eliminates the need for sharing input data with external parties, making it easier to comply with local privacy policies and regulations. Our second approach adopts the federated paradigm in which all parties jointly train a pair of generator and discriminator with differential privacy. Synthetic samples can be drawn from the global generator once training is complete.

Prior work has incorporated differential privacy in the FedAvg algorithm [20], where each user's participation in federated learning is protected. In this work, we propose to address the non-IID data challenge in federated learning with

Algorithm 1 DP-FedProx-GAN

```
Server loop:
Input: total number of parties $K \in \mathbb{N}$, total number of rounds
       $T \in \mathbb{N}$, sampling probability q $\in$ (0, 1], noise scale
       $z \in \mathbb{R}^+$ , clipping parameter $S \in \mathbb{R}^+$, generator $\theta_G^0$ ,
       discriminator $\theta_D^0$, privacy accountant $\mathcal{M}$
Initialize $\sigma = \frac{zS}{qN}$
for round $t = 0$ in $T - 1$ do
    $C^t \leftarrow$ (randomly sample $qN$ distinct parties)
    for each party $k \in C^t$ do
        $\Delta_k^{t+1} \leftarrow \text{LocalDiscUpdate}(k, \theta_D^t, \theta_G^t)$ // compute
         local update
    end
    $\Delta^{t+1} \leftarrow \frac{1}{qN} \sum_{k \in C^t} \Delta_k^{t+1}$
    $\theta_D^{t+1} \leftarrow \theta_D^t + \Delta^{t+1} + \mathcal{N}(0, I\sigma^2)$ // update
     discriminator privately
    $\mathcal{M}$.accum-privacy-spending($z$)
    $\theta_G^{t+1} \leftarrow \text{GeneratorUpdate}(\theta_D^{t+1}, \theta_G^t)$
end
$\mathcal{M}$.accum-privacy-spent() // report total privacy

LocalDiscUpdate($k, \theta_D^t, \theta_G$):
Input: batch-size $B \in \mathbb{N}$, number of disc. steps $n \in \mathbb{N}$, disc.
       learning rate $\eta_D \in \mathbb{R}^+$, clipping parameter $S \in \mathbb{R}^+$ ,
       weight for proximal term $\mu \in \mathbb{R}^+$
$\theta_D \leftarrow \theta_D^t$
$\mathcal{B} \leftarrow$ ($n$ size-$B$ batches from $D^k$ )
for each batch $b_{real} \in \mathcal{B}$ do
    $b_{fake} \leftarrow$ (sample $B$ synthetic records from generator $\theta_G$)
    $\nabla h \leftarrow \nabla h_\mu(\theta_D)$ // as in (3), parameterized
     with $\theta_D^t$, $b_{real}$, $b_{fake}$
    $\theta_D \leftarrow \theta_D - \eta_D \nabla h$ // local update
end
$\Delta = \theta_D - \theta_D^t$
return $\Delta_k = \Delta.\min(1, \frac{S}{||\Delta||})$ // clip locally

GeneratorUpdate($\theta_D, \theta_G^t$):
Input: batch-size $B \in \mathbb{N}$, number of gen. steps $n \in \mathbb{N}$, gen.
       learning rate $\eta_G \in \mathbb{R}^+$
$\theta_G \leftarrow \theta_G^t$
for each step from $i = 0$ to $n$ do
    $b_{fake} \leftarrow$ (sample $B$ synthetic records from generator $\theta_G$)
    $\theta_G \leftarrow \theta_G - \eta_G \nabla l_G(\theta_G)$ // update generator, $l_G$
     parameterized with $\theta_D$ and $b_{fake}$
end
return $\theta_G$
```

the FedProx algorithm [8] and modify the training procedure to learn GAN models with differential privacy. Algorithm 1 depicts our proposed solution, namely DP-FedProx GAN.

The advantage of DP-FedProx GAN is that it achieves user-level privacy by performing local clipping for gradient updates at each party (as in `LocalDiscUpdate`), followed by global aggregation and perturbation (as in `Server loop`). Note that it has been shown that differential privacy can be achieved by training only discriminators privately [25] (i.e., clipping and perturbations are not required for training generators), thanks to differential privacy's resistance to post-processing.

Furthermore, DP-FedProx GAN tackles the significant variability in local data by adding the proximal term to the objective. Specifically, instead of just minimizing the discriminator loss $l_D(.)$ as defined in (2), each party k employs its local procedure to minimize the following objective approximately at round t:

$$\min_{\{\theta\}} h_\mu(\theta_D) = l_D(\theta_D) + \frac{\mu}{2}||\theta_D - \theta_D^t||^2 \qquad (3)$$

The proximal term aims to address both systems and statistical heterogeneity by limiting local party updates closer to the server's global model, hence improving training stability. To assess the efficacy of our proposed solution, we compare it with the existing approach DP-FedAvg GAN [27] in the empirical evaluation.

C. Non-IID Data Simulation

Non-IID data distributions may pose significant challenges in decentralized machine learning applications [7]. Recent research has explored partitioning techniques [28] to quantify and control the imbalanced features of decentralized datasets, which is beneficial for developing and validating new machine learning algorithms. Inspired by prior research, below we discuss various types of distributions considered in this work.

Non-Skew (NS). We use the non-skew distribution as a baseline. In NS, each party has an equal-sized subset sampled uniformly at random from the overall training data.

Quantity Skew (QS). In quantity skew, the size of each party's dataset $|D^k|$ varies, without varying the label distribution. We use the Dirichlet distribution as in [29]–[32] to simulate the skewed quantities among parties. Formally, the probability density function for Dirichlet distribution is defined as follows:

$$Dir_{K,\alpha}(x) = \frac{1}{\mathrm{B}(\alpha)} \prod_{k=1}^{K} x_k^{\alpha-1} \text{ where } \mathrm{B}(\alpha) = \frac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma(\sum_{i=k}^{K}\alpha_k)} \qquad (4)$$

in which x is a K-dimensional variable and $\alpha = (\alpha_1, ..., \alpha_K)$ representation concentration parameters. For simplicity, we assume the same concentration parameter for every k. To simulate different data quantifies across K parties, we draw $q \sim Dir_{K,\alpha}$ and assign q_k proportion of training data to D^k. We can control the amount of quantity imbalance by varying the α value. A smaller α leads to a higher imbalance and vice versa, as shown in Fig. 2.

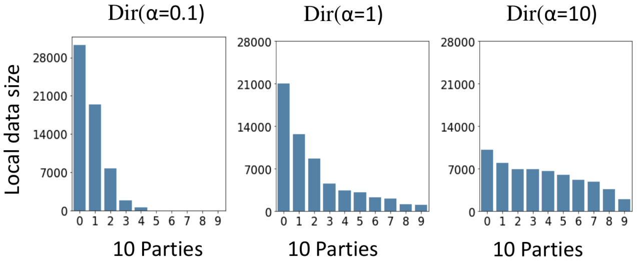


Fig. 2: Examples of Quantity Skew: simulated 10 parties with the MNIST dataset by varying α in Dirichlet distribution.

Quantity-based Label Imbalance (QLI). Besides quantity, the label distribution may vary from one party to another. Some studies [18], [29] noted that local data may contain different subsets of labels. For example, one party may have samples from classes 1 and 2, while another may have samples from classes 2 and 3. To simulate quantity-based label imbalance, we assign 3 classes to each party and randomly assign samples of those classes to the party as in [32]. Note that the quantities of local data are balanced in this setting.

Distribution-based Label Imbalance (DLI). In distribution-based label imbalance, samples of each label is distributed non-IID across the parties. For each label j, we draw $p^j \sim Dir_{K,\alpha}$ and assign p_k^j proportion of samples of label j to party k as described in [32]. Note that in DLI, both the quantity and the label distribution of the local dataset may differ between one party and another.

V. Evaluation Measures of Data Synthesis

In this section, we describe a number of quality and privacy measures adopted in evaluating data synthesis solutions. As a proof of concept, this study focuses on image data, while the proposed private data synthesis solutions can be generalized to other domains.

Utility Metrics. To simulate human evaluation of synthetic samples, strategies for assessing the realism and diversity of generated data have been developed for GANs. Among them, Inception Score (IS) [33], and Fréchet Inception Distance (FID) [34] have been extensively evaluated in the literature. The IS captures the KL divergence between the conditional label distribution (estimated with the Inception model) and the marginal distribution for synthetic samples. The FID computes the Fréchet distance between the distributions of real data and generated data.

Although the IS and FID scores take into account the unambiguity and relative abundance of distinct classes in generated data, we look at those measures explicitly via entropy and diversity measures as in [26]. Entropy is calculated for the conditional label distribution of synthetic samples, which is also estimated using the inception model. The diversity of synthetic samples is computed using the most likely label for each sample in the conditional distribution.

Membership Inference Risks. As the proposed solutions adopt various privacy notions, i.e., non-differential privacy, record-level DP, and user-level DP, it is important to evaluate their privacy protection via common empirical measures. An important class of privacy attacks in machine learning is membership inference [6], [10]. With access to the learned GAN models, the following membership inference attacks may be launched. Although model sharing is not necessary in data synthesis, evaluating those attacks provides insights into the privacy properties of learned models and helps assess the privacy loss in worst cases, e.g., by model stealing.

We adopt two attacks on GAN discriminators to the decentralized setting, i.e., the white-box (WB) attack [5] and the total variation distance (TVD) attack [26]. In these attacks, an adversary aims to infer whether a sample is used in training the GAN models and does not know the source of the sample, i.e., the party holding the sample. When GAN models are trained locally, we adapt the original WB and TVD attacks such that the adversary takes the maximum discriminator score over all parties, with the intuition that the discriminator trained with a specific sample will have the highest discriminator score.

The generator in GANs is also prone to set membership inference in Monte–Carlo (MC) attacks [35]. In the MC attack, the adversary's objective is to determine whether a given set of samples are part of the GAN models' training set or synthetically generated. It computes the similarity between samples in each set using PCA transformation and preserves 40 principal components. The MC attack can be readily evaluated using synthetic samples drawn from locally trained GAN models as well as federated GAN models.

VI. Experiments

In this section, we present the evaluation methodology and empirical results on the proposed data synthesis solutions.

Datasets. Our experiments adopt the following datasets: MNIST, Fashion-MNIST(f-MNIST), CIFAR-10, and CelebA. Note that to create a balanced dataset, we sub-sampled the original CelebA [36] to obtain 30 images per identity for 1000 celebrities. We also center-cropped each image to obtain face regions of 48 × 48 pixels.

Approaches. We evaluate several approaches, i.e., local training of privGAN and DPGAN models and our proposed DP-FedProx GAN. In addition, we include DP-FedAvg GAN [27], which is the current solution for training GAN models in decentralized settings. It is important to note that we also consider an alternative approach, namely DP-FedSGD GAN, which learns federated GAN models using the DP-FedSGD algorithm [20]. However, we omit DP-FedSGD GAN from our evaluation, as it requires a higher number of training iterations and performs poorly compared to the other approaches.

Default Parameter Values. λ and ϵ parameters specify the degree of privacy protection in privGAN, DPGAN, and DP-FedProx GAN. The default values used are λ=1 in privGAN, ϵ=4.01 in DPGAN, and ϵ=64.11 in DP-FedProx GAN and DP-FedAvg GAN. The default sampling probability q is set to 0.1 in DP-FedProx GAN and DP-FedAvg GAN. For DP-FedProx GAN, μ in (3) specifies the weight of the proximal term and is set to 0.5. The concentration parameter α indicates the level of imbalance, and the default value is set $\alpha = 0.5$. The number of parties K is set as $K = 10$, unless specified otherwise. Table I presents clipping parameters S for differentially private solutions in each dataset.

	MNIST	f-MNIST	CIFAR-10	CelebA
DPGAN	1e-2	1e-2	5e-2	5e-2
DP-FedProx GAN	1e-2	1e-2	5e-2	5e-2
DP-FedAvg GAN	1e-2	5e-2	5e-2	5e-2

TABLE I: Clipping Parameters for Differentially Private Solutions

		MNIST	f-MNIST	CIFAR-10	CelebA
Real Data	IS	9.9	9.6	9.1	141.4
Centralized GAN ($K = 1$)	IS	9.2	9.0	7.8	88.8
	FID	12.2	14.2	30.1	45.1
Local GANs ($K = 10$)	IS	5.5	5.20	4.12	28.30
	FID	40.80	38.50	80.23	110.54
Federated GAN ($K = 10$)	IS	8.4	8.61	5.50	39.41
	FID	16.21	25.50	56.41	78.45

TABLE II: IS and FID Scores: Real Data, Centralized Non-Private GAN ($K = 1$), Non-Private Local and Federated GANs with NS distribution ($K = 10$).

A. Varying privacy parameters

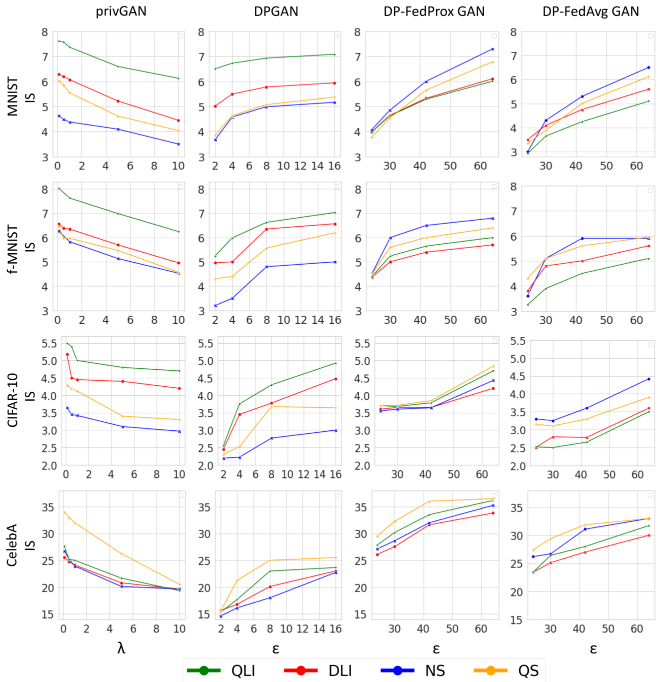


Fig. 3: Varying Privacy Parameters

This experiment shows how the privacy parameters of private GANs affect the utility metrics. As a reference, we show IS and FID scores for non-private centralized GAN as well as non-private local and federated GAN models in Table II. In comparison to real data and synthetic data generated by centralized GAN, all decentralized GAN solutions lead to high utility loss, i.e., much lower IS scores. It can be seen that GAN

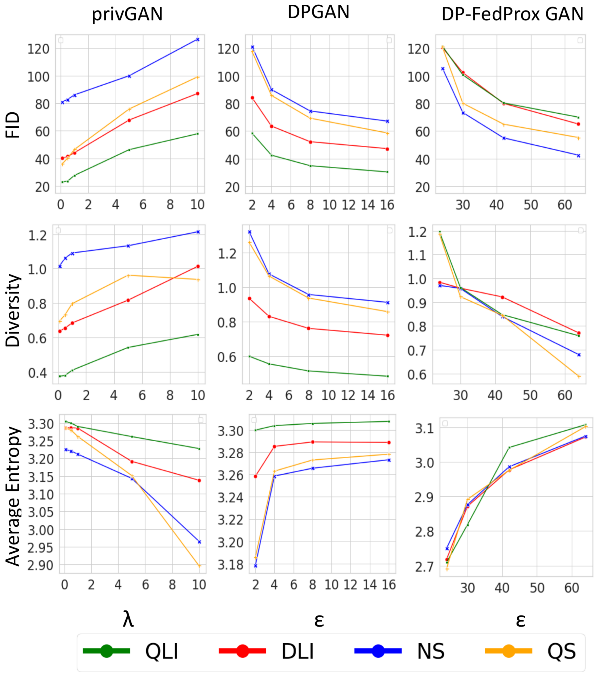


Fig. 4: Class Diversity and Average Entropy with MNIST

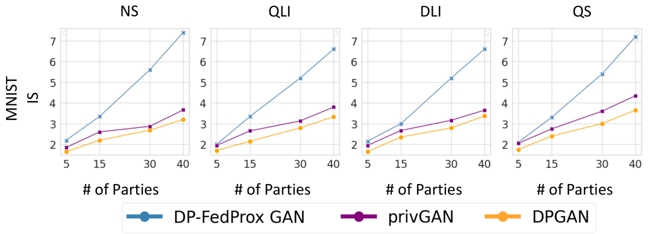


Fig. 5: Varying Number of Parties with MNIST

models trained locally suffer from limited data at each party. As CelebA is smaller in size and higher in resolution, it is more challenging to generate high-quality images in decentralized settings.

Fig. 3 shows the results of IS measures for all private solutions and all datasets. When we increase λ in privGAN and decrease ϵ in DPGAN, DP-FedProx GAN and DP-FedAvg GAN, the utility improves monotonically. We observe that training GANs locally (i.e., privGAN and DPGAN) does not learn the underlying distribution of local data well in non-skew (NS) settings. The reason is that none of the local parties holds sufficient data. For example, NS distribution has the lowest utility across all λ in privGAN and ϵ in DPGAN. However, DP-FedProx GAN shows a higher utility with NS, in comparison to non-IID distributions. As a random subset of local parties participates in each round of training the DP-FedProx GAN, NS distributions can be learned uniformly across parties. We note that DP-FedProx GAN effectively manages statistical heterogeneity by constraining local updates to be closer to the global model, as all non-IID distributions yield similar performance to that of NS. As expected, DP-FedAvg GAN struggles to learn from non-IID data distributions and consistently yields lower utility than DP-FedProx GAN across all datasets. Therefore, we omit DP-FedAvg GAN from the rest of the experiments.

In QLI settings, locally trained GAN solutions often lead to good utility because each party focuses on learning only a few classes without encountering mode collapse and instability in GAN training. For instance, QLI has the highest IS for privGAN and DPGAN across all λ and ϵ values, respectively, in MNIST, f-MNIST, and CIFAR-10. In QS settings, we observe that parties with more data are comparatively more influential (i.e., better utility) for privGAN and DPGAN solutions. QS also leads to better utility in DP-FedProx GAN for the dataset, in comparison to other distributions.

In Fig. 4, we report the effects of privacy parameters on FID score, class diversity, and average entropy with the MNIST dataset. As λ increases in privGAN and ϵ decreases in DPGAN and DP-FedAvg GAN, both FID and entropy increase, and diversity decreases accordingly. QLI shows the highest class diversity and the lowest FID and average entropy for PrivGAN and DPGAN solutions. Again, all distributions yield similar performances in DP-FedProx GAN, indicating its efficacy in addressing non-IID distributions.

B. Varying the Number of Parties

We investigate the effects of the number of parties (K) on utility metrics with the MNIST dataset, as shown in Fig. 5. To conduct this experiment, we partition the training set among $K = 40$ parties in each distribution setting and select the specified number of parties for model training. We observe higher utility as K increases because private GANs utilize more data contributed by a larger number of parties, which is beneficial for generating high-quality and diversified synthetic data. DP-FedProx GAN consistently outperforms privGAN and DPGAN, as locally trained GAN models perform poorly due to small amounts of local data. When larger numbers of parties are involved ($K > 15$), a significant increase in utility can be observed for DP-FedProx GAN, as opposed to privGAN and DPGAN. At $K = 40$, DP-FedProx GAN achieves around 7 Inception Score while that of privGAN and DPGAN is around 4. It illustrates that the DP-FedProx GAN approach leads to the higher utility when a large number of parties participate.

C. Varying Concentration Parameter for Skew Distribution

In this experiment, we investigate how private data synthesis solutions behave by varying the concentration parameter α in Fig. 6. Recall that smaller α values lead to more unbalanced distributions. As we increase the α, the data distributions approach the NS setting; as a result, utility drops for privGAN and DPGAN and increases for DP-FedProx GAN. This result is consistent with our observations early on.

When data is highly imbalanced, i.e., low α values, locally trained GANs perform better in DLI than in QS. The reason is that highly imbalanced label distributions lead to samples from

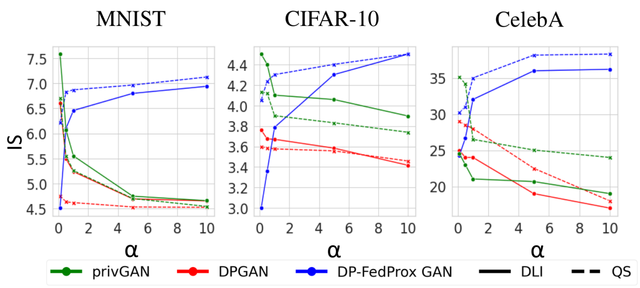


Fig. 6: Varying Concentration Parameter

		WB	MC	TVD
Centralized GAN ($K = 1$)	-	48.1	77.1	0.438
Local GANs ($K = 10$)	NS	11.50	68.8	0.37
	QLI	25.10	72.1	0.45
	DLI	17.50	69.3	0.41
	QS	18.25	70.5	0.44
Federated GAN ($K = 10$)	NS	21.50	72.8	0.44
	QLI	20.00	71.1	0.41
	DLI	20.50	68.1	0.40
	QS	21.12	72.2	0.43

TABLE III: Attack results on synthetic images generated with non-private centralized private GAN ($K = 1$) as well as non-private local GANs and federated GAN model ($K = 10$) with MNIST dataset.

fewer classes at each party, which resembles the QLI setting. From our observations early on, we know that privGAN and DPGAN perform best in the QLI setting. As α increases, the difference between DLI and QS diminishes, so does the performance gap between those settings for privGAN and DPGAN. With lower α values, DP-FedProx GAN performs better in QS than in DLI, as each party in QS settings has a balanced label distribution, which is more helpful for the federated solution.

D. Membership Inference Attacks

In this experiment, we vary the privacy parameters for the private data synthesis solutions to study their defense against membership inference attacks (i.e., WB, MC, and TVD) with the MNIST dataset. Details regarding the attack implementations can be found in the appendix. As a reference, in Table III we present the attack results for non-private centralized GAN as well as local and federated GAN models. In decentralized settings, we observe consistent reductions in the accuracy of WB and MC attacks compared to the centralized setting, but increased TVD scores in some distributions.

In Fig. 7, we report the attack accuracies for private solutions by varying λ in privGAN and ϵ in DPGAN and DP-FedProx GAN. We observe that private solutions reduce the accuracy of WB, MC, and TVD attacks consistently, with respect to their non-private baselines in decentralized settings. Furthermore, as we improve the privacy protection, i.e., increasing λ in privGAN and decreasing ϵ in DPGAN and DP-FedProx GAN, the attack accuracies can be further reduced.

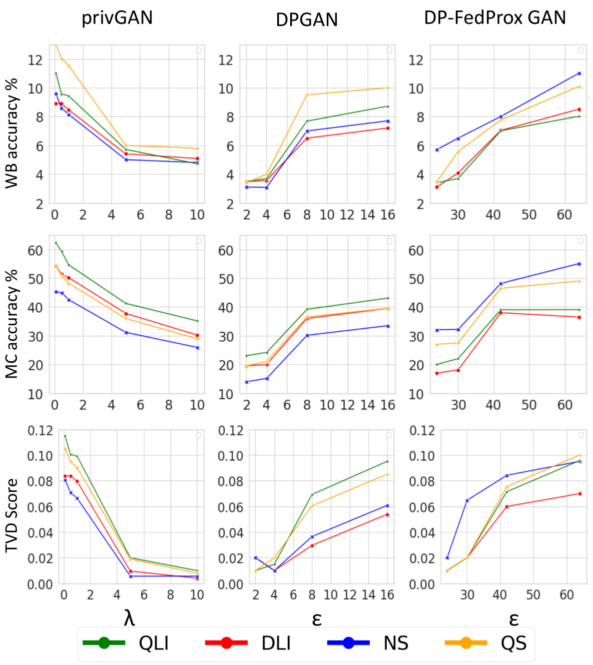


Fig. 7: Attacks on Discriminators and Generators with MNIST

For locally trained private GANs (privGAN and DPGAN), privacy attacks in the NS setting may be the least successful, compared to non-IID settings. The lower privacy risk corresponds to the lower utility for privGAN and DPGAN in NS as in Fig. 3, as private local GAN models struggle to learn from small local data. However, in DP-FedProx GAN, the NS distribution is among the most susceptible to membership attacks, as the federated GAN model learns data distributions well from all parties.

In the QLI setting, as privGAN and DPGAN perform relatively well in utility (see Fig. 3), the learned generators and discriminators are more prone to membership inference attacks, e.g., higher MC and TVD scores in Fig. 7. We also observe that the privGAN and DPGAN discriminators learned in the QS setting can leak membership information, i.e., resulting in higher WB accuracy than other settings, despite lower utility in data synthesis. On the other hand, in DP-FedProx GAN, the learned discriminator and generator are consistent in privacy leakage among all distribution settings. Furthermore, the privacy leakage is also consistent with DP-FedProx GAN utility results among all distribution settings. In the next subsection, we further examine the trade-off between empirical privacy and utility for all private solutions.

E. Privacy vs. Utility

We note that the proposed data synthesis solutions do not share the privacy notion. Specifically, privGAN doesn't provide any privacy guarantees, while DPGAN and DP-FedProx GAN provide record-level and user-level differential privacy guarantees, respectively. Therefore, we conduct a privacy utility trade-off analysis among those solutions in terms of WB membership inference risks in Fig. 8. It can be observed

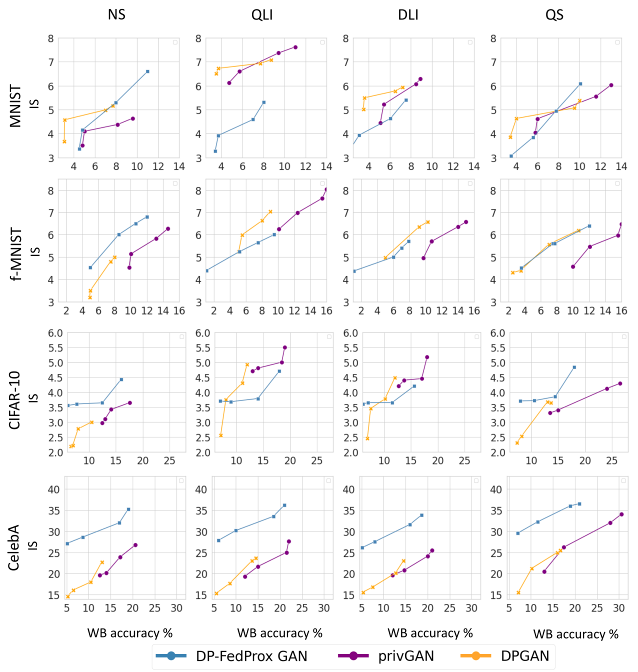


Fig. 8: Privacy vs. Utility Tradeoff

that generally privGAN results in higher WB attack risks than DPGAN and DP-FedProx GAN.

We observe that DP-FedProx GAN the best privacy-utility tradeoff in NS distributions: given a WB accuracy, DP-FedProx GAN often provides better IS scores than privGAN and DPGAN. DPGAN provides good privacy-utility trade-off in non-IID distributions for three simpler datasets (i.e., MNIST, f-MNIST, CIFAR-10). In non-IID distributions for those datasets, privGAN may provide better utility at the expense of increased privacy risks.

For the CelebA dataset, DP-FedProx GAN clearly outperforms privGAN and DPGAN, by consistently dominating two other methods in utility at the same privacy risk level. This shows DP-FedProx GAN is superior in balancing privacy and utility with a large number of classes (1000) and few samples per class (30). Both locally trained private GANs suffer from lower utility and privGAN yields higher privacy risks, due to their lack of privacy guarantees.

VII. Discussion

Interpreting the Results. Considering the results combined, we note that DP-FedProx GAN outperforms privGAN and DPGAN in non-skew and moderately skewed settings (see Fig. 3 and 6). In highly skewed settings, its strong user-level differential privacy may lead to higher utility loss, as parties have equal chance of participating in each round despite the amount of local data. On the other hand, training privGAN or DPGAN locally may yield good utility in non-IID settings, as parties with larger amounts of data dominate the quality measures. Among different non-IID distributions, local privGAN and DPGAN show great promise for QLI, where each party focuses on synthesizing samples of a small number of classes; DP-FedProx GAN produces similar results for all non-IID distributions, demonstrating its capability to learn a variety of data distributions. For complex tasks, e.g., learning the subsampled CelebA with a large number of classes and fewer samples per class, all approaches produce better quality data in QS settings; the difference between distributions diminishes by increasing the amount of skewness as in Fig. 6. Lastly, increasing the number of parties significantly boosts the utility of DP-FedProx GAN, while privGAN and DPGAN only show marginal utility gain as in Fig. 5.

Deployment Choice. Given different privacy models and wide variability in utility performances, there is no one-size-fits-all solution for private synthesis with decentralized non-IID data. While training GAN models locally provides full autonomy to each party, training federated GAN models allow all parties to contribute meaningfully, even those with very small amounts of local data. Furthermore, DP-FedProx GAN motivates local parties to jointly train models, as it shows significant utility benefits as the number of parties increases in Fig. 5. Moreover, we showcase how to conduct a trade-off analysis between utility measures and empirical privacy risks as in Fig. 8. It can be observed that privGAN may lead to higher privacy risks, due to lack of a formal privacy model. We argue that approaches that provide differential privacy guarantees (i.e., DPGAN and DP-FedProx GAN) should be considered due to their rigorous and future-proof privacy protection.

Practical Considerations. Needless to say, a number of considerations should be addressed for the practical adoption of the proposed solutions. Firstly, all parties must agree on the training mode (i.e., local vs. federated) and the privacy settings (e.g., λ and ϵ). When parties have different privacy requirements (e.g., in terms of ϵ values for DP), the strongest privacy level (e.g., lowest ϵ) should be adopted to ensure privacy for all parties. Secondly, in the local GAN training approach, when each party contributes synthetic data of the same size as that of the local dataset, it may disclose aggregate-level information about private data. That may be addressed by applying differential privacy (e.g., Laplace mechanism [15]) to the local count and sampling synthetic records according to the noise count. Lastly, we recommend grid search approaches for uncovering suitable hyper-parameters in local training approaches and weight-sharing for the federated GAN training. Recent research on hyper-parameter optimization in federated settings [37] may provide new opportunities for the practical deployment of the federated approach.

VIII. Conclusion

In this paper, we studied several practical solutions for private synthesis using GANs with decentralized, non-IID data. Among them, privGAN and DPGAN can be trained by each party locally and DP-FedProx GAN is trained jointly by all parties with strong user-level differential privacy guarantees. We conducted an extensive empirical evaluation with data from multiple image domains and simulated a variety of non-IID distributions. We provided an in-depth analysis of the evaluation results, regarding the usefulness of synthetic data, privacy risks in membership inference attacks, and the privacy utility trade-off for the proposed solutions.

Several directions are open for future research on private data synthesis from decentralized non-IID data. Firstly, it is helpful for future research to address data with imbalanced labels. Privacy risks may be higher due to class imbalance [38]. Achieving high data synthesis utility for class-imbalanced data may also be challenging, especially for rare classes. Secondly, understanding the usefulness of synthetic data in various applications would be beneficial, e.g., in clinical decision support systems. Thirdly, future research may extend to emerging federated learning results for non-IID data, such as FedCurv [39], which proposes to share additional elements by each local party to overcome forgetting. As a result, such approaches may lead to additional privacy risks and thus demand new solutions. Lastly, it would be interesting to develop grid search approaches for hyper-parameters in federated GAN training, with privacy guarantees, e.g., differential privacy. It will help provide end-to-end privacy for data synthesis in decentralized settings.

Acknowledgment

The authors would like to thank the anonymous reviewers for their feedback, which helped improve the manuscript.

References

[1] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, p. 557–570, oct 2002.

[2] M. Barbaro and T. Zeller, "A face is exposed for aol searcher no. 4417749," *New York Times*, 01 2006.

[3] Y.-A. Montjoye, C. Hidalgo, M. Verleysen, and V. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific reports*, vol. 3, p. 1376, 03 2013.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, oct 2020.

[5] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "Logan: Membership inference attacks against generative models," 2018.

[6] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020, pp. 343–362.

[7] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4387–4398.

[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018.

[9] D. Saxena and J. Cao, "Generative adversarial networks (gans): Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 54, no. 3, may 2021.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[11] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333.

[12] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.

[13] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1175–1191.

[14] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[15] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, p. 211–407, Aug. 2014.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318.

[17] L. Fan, "A survey of differentially private generative adversarial networks," in *The AAAI Workshop on Privacy-Preserving Artificial Intelligence*, 2020, p. 8.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282.

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

[20] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.

[21] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, aug 2017.

[22] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled renyi differential privacy and analytical moments accountant," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 1226–1235.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[25] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[26] S. Mukherjee, Y. Xu, A. Trivedi, and J. L. Ferres, "privgan: Protecting gans from membership inference attacks at low cost," 2020.

[27] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. A. y. Arcas, "Generative models for effective ml on private, decentralized datasets," 2019.

[28] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[29] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7252–7261.

[30] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[31] Q. Li, B. He, and D. Song, "Practical one-shot federated learning for cross-silo setting," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1484–1490, main Track.

[32] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 965–978.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[34] 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," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[35] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models," vol. 2019, 07 2019.

[36] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[37] M. Khodak, R. Tu, T. Li, L. Li, M.-F. F. Balcan, V. Smith, and A. Talwalkar, "Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 184–19 197, 2021.

[38] T. Farrand, F. Mireshghallah, S. Singh, and A. Trask, "Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy," in *PPMLP'20: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, Virtual Event, USA, November, 2020*, B. Zhang, R. A. Popa, M. Zaharia, G. Gu, and S. Ji, Eds. ACM, 2020, pp. 15–19.

[39] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak, "Overcoming forgetting in federated learning on non-iid data," *arXiv preprint arXiv:1910.07796*, 2019.

[40] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

APPENDIX

A. Implementation Details

For MNIST and f-MNIST, we use standard, fully connected networks for both generators and discriminators. On the other hand, we adopt a Deep Convolutional Generative Adversarial Network (DCGAN) structure for EMNIST [40], CIFAR-10, and CelebA datasets. All GAN models have identical generator and discriminator architectures for the respective dataset (see details below).

For evaluation, we train the privGAN models with an Adam (β=5) optimizer for both the generator and discriminator. For differentially private GANs (i.e., DPGAN, DP-FedProx GAN, DP-FedAvg GAN, and DP-FedSGD GAN), we use Differentially Private Stochastic Gradient Descent (DP-SGD) optimizer for the discriminator. For the generator, we use Adam (β=5) optimizer in DPGAN and SGD optimizer in DP-FedProx GAN, DP-FedAvg GAN, and DP-FedSGD GAN. We set a 0.0002 learning rate for all optimizers. The batch size adopted for privGAN is 256. For DP-GAN, the batch size is varied from 16 to 64 in order to meet the specified privacy parameter ϵ. For DP-FedProx GAN and DP-FedAvg GAN, the batch size is set to 32 for CIFAR-10 and CelebA and to 64 for other datasets. For each round in DP-FedSGD GAN, we randomly select a batch size of qD^k for the k-th party, where q signifies the sampling probability (0.1) and D^k refers to the local data size of the k-th party. The default value for DP-FedSGD GAN is set at ϵ=98.01.

While evaluating the different adversarial attacks on PrivGAN, we train PrivGAN for 500 epochs using the same optimizer and hyperparameters. Similarly, we train all differentially private GAN models with a fixed privacy budget ϵ, setting the noise scale z to achieve the specified ϵ. For WB and TVD attacks, we use 10% of the training set to train models, following the approach in [5].

To evaluate the MC attack, we follow the methodology in [26], [35]. We use 10% of the training set for training the attack model and evaluate the model on the residual 90% of the training set. The test set is utilized solely to calculate the principal components for all datasets. In federated training (i.e., DP-FedProx GAN, DP-FedAvg GAN, and DP-FedSGD GAN), the central server generates 100,000 synthetic samples. In local training (i.e., PrivGAN and DPGAN), we sample 100,000 synthetic samples from each party.

In order to ensure the results are representative, we performed each experiment five times and reported the average outcomes for both utility and adversarial assessments.

B. Complexity Analysis

In DP-FedProx GAN and DP-FedAvg GAN, the time complexity for each party is $O(nBT))$, where n represents the number of steps for local discriminator and generator updates, B represents the batch size, and T represents the total number of rounds. In the context of privGAN and DPGAN, the time complexity for each party can be expressed as $O(DE)$, where D is the upper bound of local data size, and E corresponds to the number of epochs for the local model training.

◇ **MNIST and f-MNIST**

Generator Layers

- Dense(units= 256, input size= 100)
- LeakyReLU(α = 0.2)
- Dense(units= 512)
- LeakyReLU(α = 0.2)
- Dense(units= 1024)
- LeakyReLU(α = 0.2)
- Dense(units= 784, activation = 'tanh')

Discriminator Layers

- Dense(units= 1024)
- LeakyReLU(α = 0.2)
- Dense(units= 512)
- LeakyReLU(α = 0.2)
- Dense(units= 256)
- LeakyReLU(α = 0.2)
- Dense(units= 1, activation = 'sigmoid')

Privacy Discriminator Layers in privGAN

- Dense(units= 1024)
- LeakyReLU(α = 0.2)
- Dense(units= 512)
- LeakyReLU(α = 0.2)
- Dense(units= 256)

- LeakyReLU(α = 0.2)
- Dense(units = # generators, activation ='softmax')

◇ **CIFAR-10**

Generator Layers

- Dense(units= 2048, input size= 100, target shape= (2, 2, 512))
- Conv2DTranspose(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 64, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 3, kernel size= 5, strides= 2, activation = 'tanh')

Discriminator Layers

- Conv2D(filters= 64, kernel size= 5, strides= 2)
- Reshape(target shape= (2, 2, 512))
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Dense(units= 1, activation = 'sigmoid')

Privacy Discriminator Layers in privGAN

- Conv2D(filters= 64, kernel size= 5, strides= 2)
- Reshape(target shape= (2, 2, 512))
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Dense(units = # generators, activation = 'softmax')

◇ **CelebA**

Generator Layers

- Dense(units= 2048, input size= 100, target shape= (2, 2, 512))
- Conv2DTranspose(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 64, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2DTranspose(filters= 3, kernel size= 5, strides= 3, activation = 'tanh')

Discriminator Layers

- Conv2D(filters= 64, kernel size= 5, strides= 2)
- Reshape(target shape= (2, 2, 512))
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Dense(units= 1, activation = 'sigmoid')

Privacy Discriminator Layers in privGAN

- Conv2D(filters= 64, kernel size= 5, strides= 2)

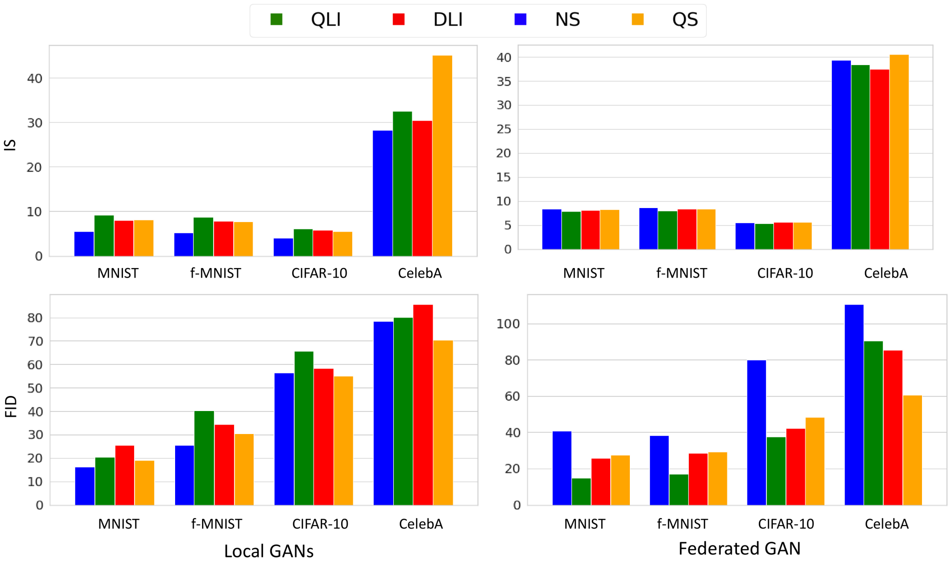


Fig. 9: Inception Score with Non-Private Local GANs and Federated GAN: The Figure shows the inception score and FID score for different distributions for each dataset in local GANs and federated GAN with $K = 10$.

- Reshape(target shape= (2, 2, 512))
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 128, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Conv2D(filters= 256, kernel size= 5, strides= 2)
- LeakyReLU(α = 0.2)
- Dense(units = # generators, activation = ’softmax’)

◇ **EMNIST**

Generator Layers

- Dense(units= 1024, input size= 100)
- Dense(units= 7 * 7 * 256)
- Reshape(target shape= (7, 7, 256))
- Conv2D(filters= 64, kernel size= 4, strides= 2)
- BatchNormalization()
- LeakyReLU(α = 0.01)
- Conv2D(filters= 32, kernel size= 4, strides= 2)
- BatchNormalization()
- LeakyReLU(α = 0.01)
- Conv2D(filters= 1, kernel size= 4)

Discriminator Layers

- Conv2D(filters= 64, kernel size= 4, strides= 2)
- LeakyReLU(α = 0.01)
- Conv2D(filters= 128, kernel size= 4, strides= 2)
- LeakyReLU(α = 0.01)
- Flatten()
- Dense(units= 1024)
- LeakyReLU(α = 0.01)
- Dense(units= 1, activation = ’sigmoid’)

C. Additional Experiments on Utility and Membership Inference Attacks

For all following experiments, we use the default parameter values listed in Section 6 unless otherwise stated.

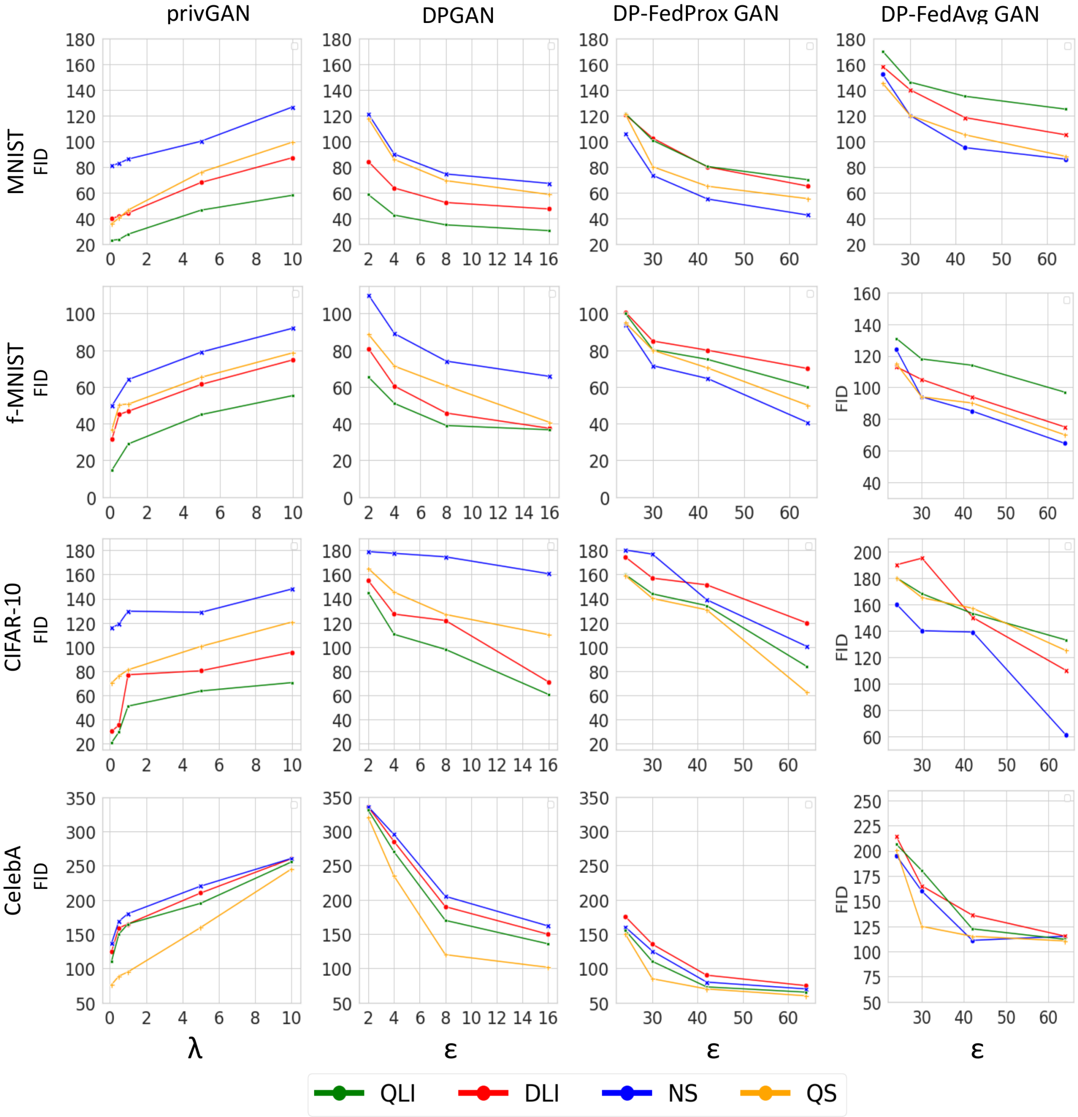


Fig. 10: Varying Privacy Parameters: The Figure illustrates FID Score for various privacy parameters in privGAN, DPGAN, DP-FedProx GAN, and DP-FedAvg GAN.

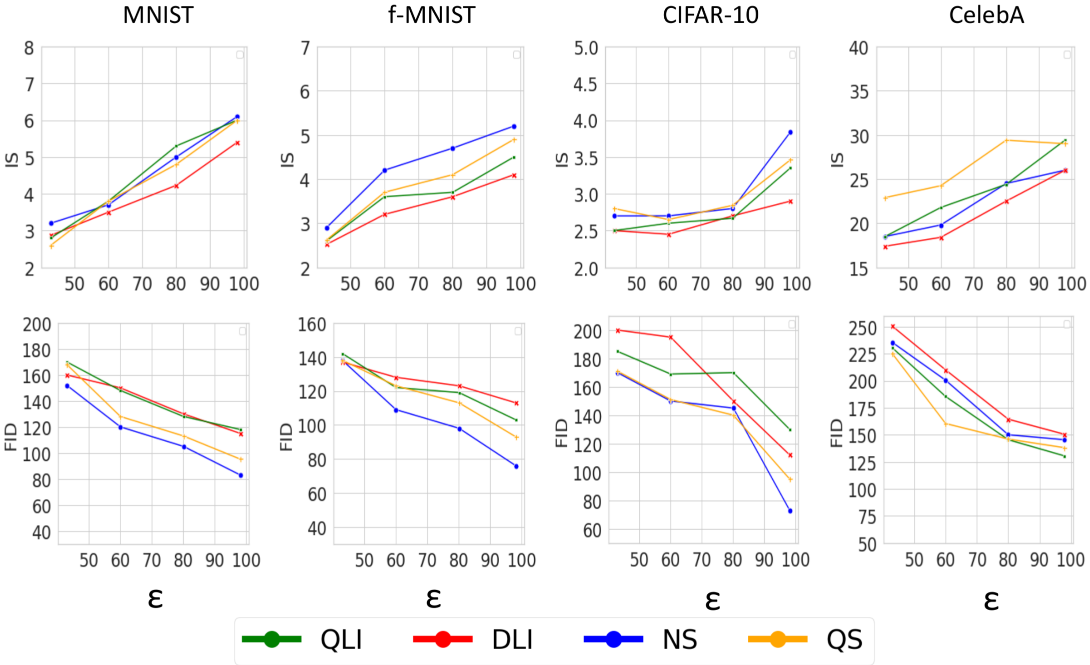


Fig. 11: Varying Privacy Parameters in DP-FedSGD GAN: The Figure represents inception and FID Score at various privacy parameters in DP-FedSGD GAN. DP-FedSGD GAN has the worst utility in terms of IS and FID compared to the DP-FedProx GAN and DP-FedAvg GAN, as shown in the Fig 3 and 10.

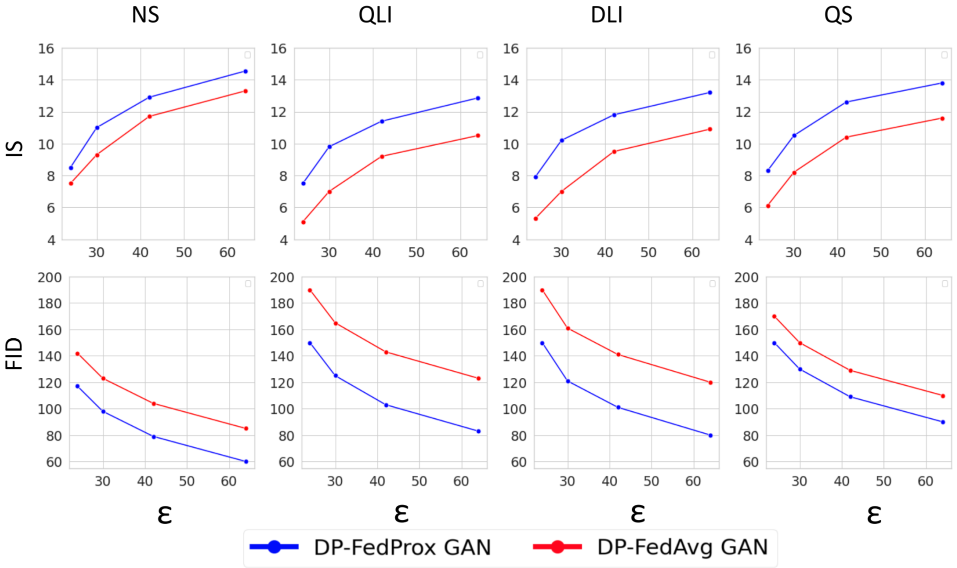


Fig. 12: Varying Privacy Parameters on EMNIST. The Figure depicts the inception and FID score at various privacy parameters in DP-FedProx GAN and DP-FedAvg GAN at K=500 on EMNIST.

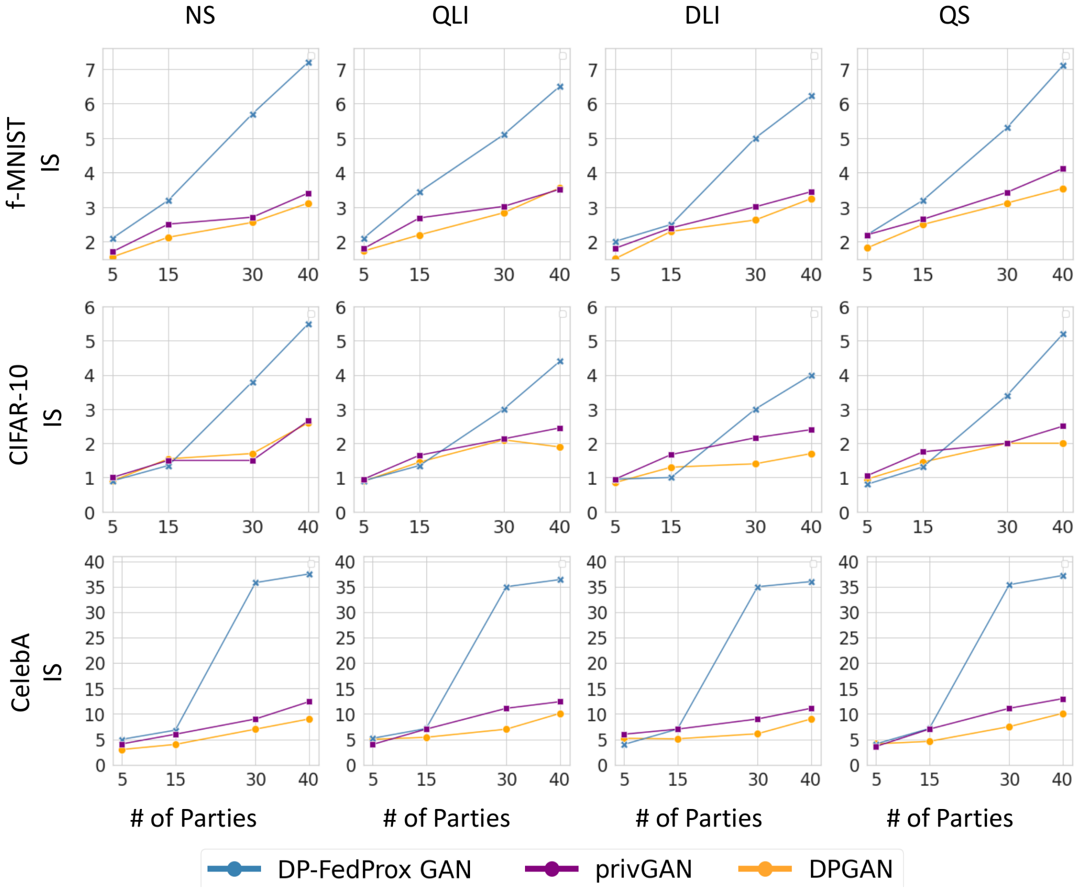


Fig. 13: Varying Number of Parties: The Figure depicts the inception score for various distributions at various numbers of parties in privGAN, DPGAN, and DP-FedProx GAN.

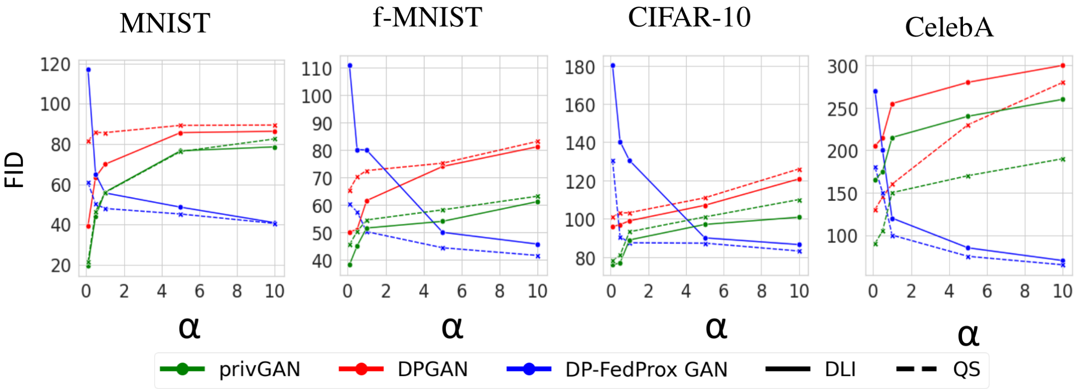


Fig. 14: Varying Concentration Parameters: The Figure illustrates the FID score for different distributions at various concentration parameters in privGAN, DPGAN, and DP-FedProx GAN.

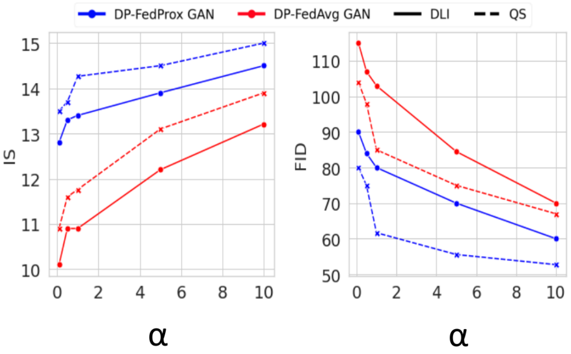


Fig. 15: Varying Concentration Parameter on EMNIST with 500 Parties (K=500) using DP-FedProx GAN and DP-FedAvg GAN .

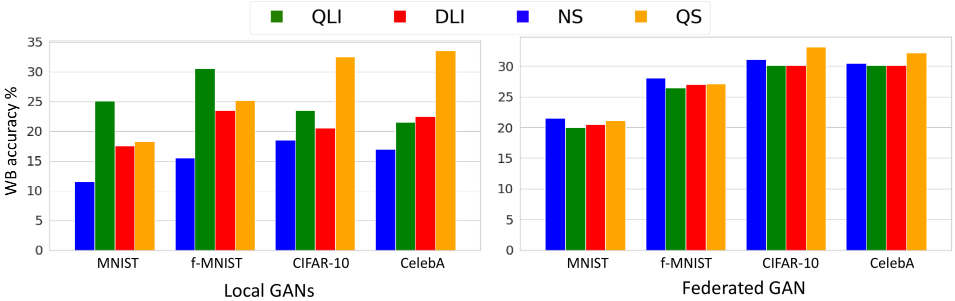


Fig. 16: White-Box accuracy for Non-Private Local GANs and Federated GAN: The Figure displays the results of a white-box attack using non-private local GANs and federated GAN with $K = 10$ for various distributions for each dataset.

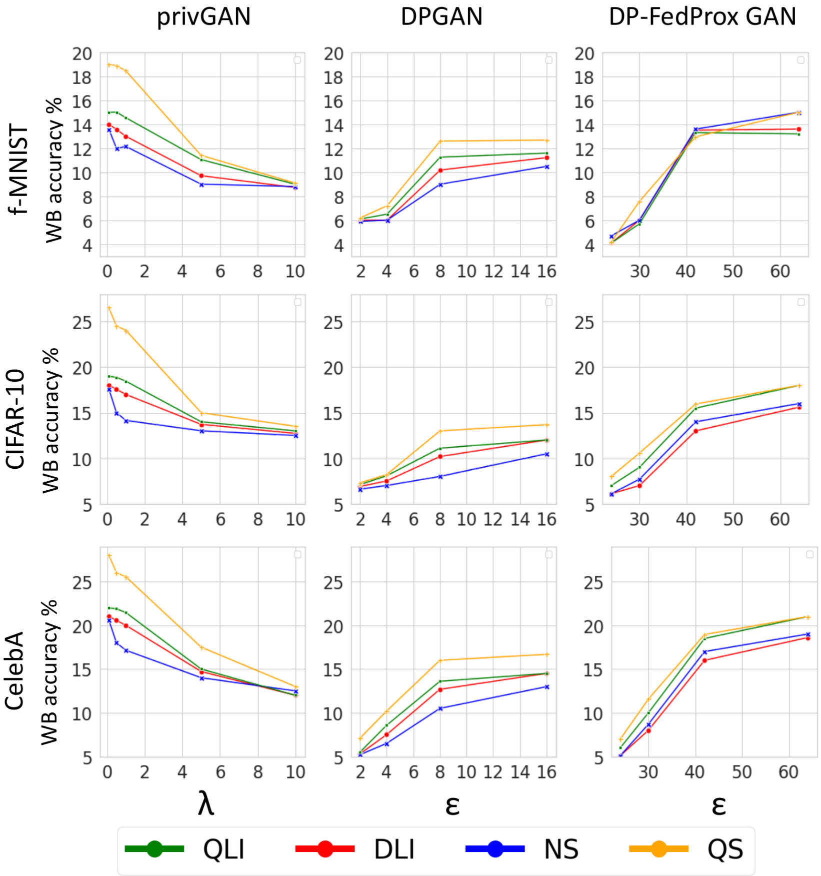


Fig. 17: Varying privacy parameters for White-Box Attack: The Figure shows white-box accuracy on discriminator(s) in privGAN, DPGAN, and DP-FedProx GAN for various privacy parameters.

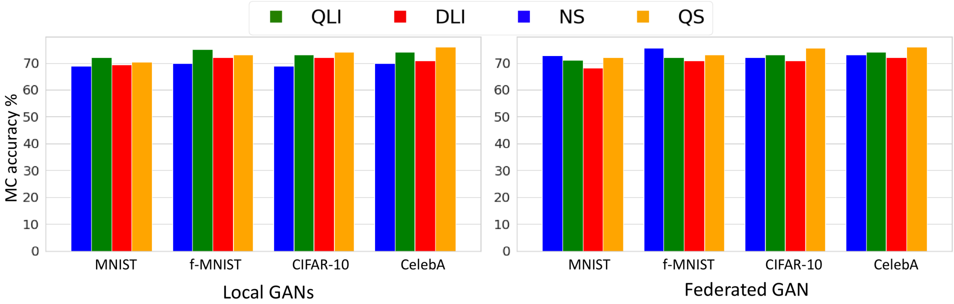


Fig. 18: Monte-Carlo accuracy for Non-Private Local GANs and Federated GAN: The Figure displays the results of a monte-carlo attack using non-private local GANs and federated GAN with $K = 10$ for various distributions for each dataset.

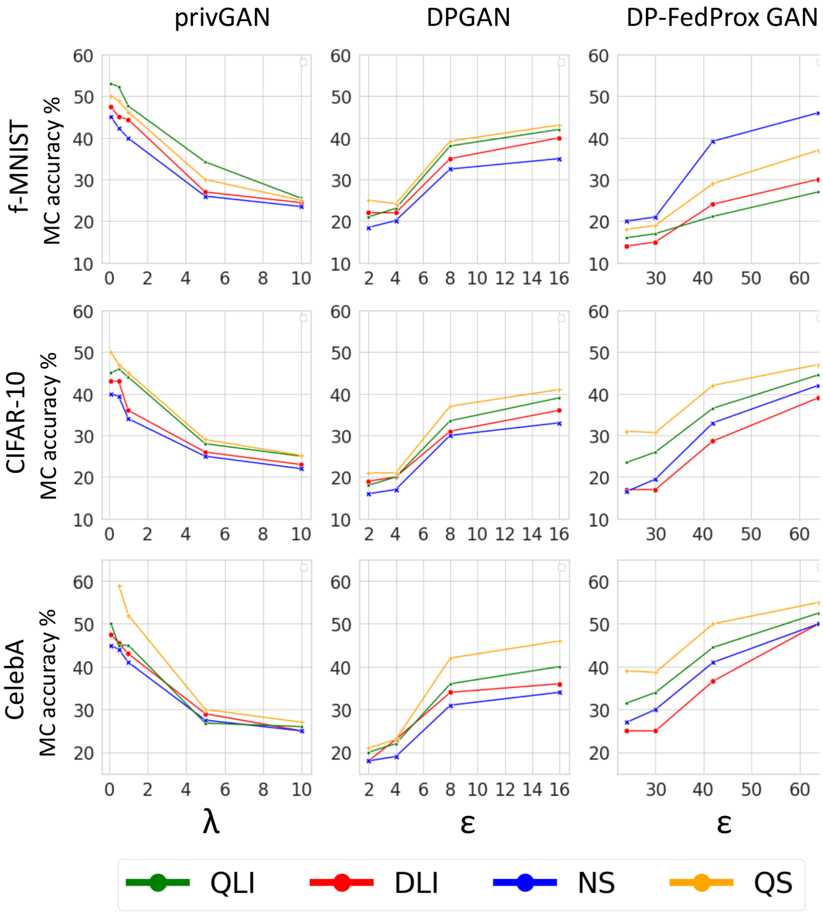


Fig. 19: Varying privacy parameters for Monte-Carlo Attack: The Figure shows Monte-Carlo Attack on the generator(s) in privGAN, DPGAN, and DP-FedProx GAN for various privacy parameters.

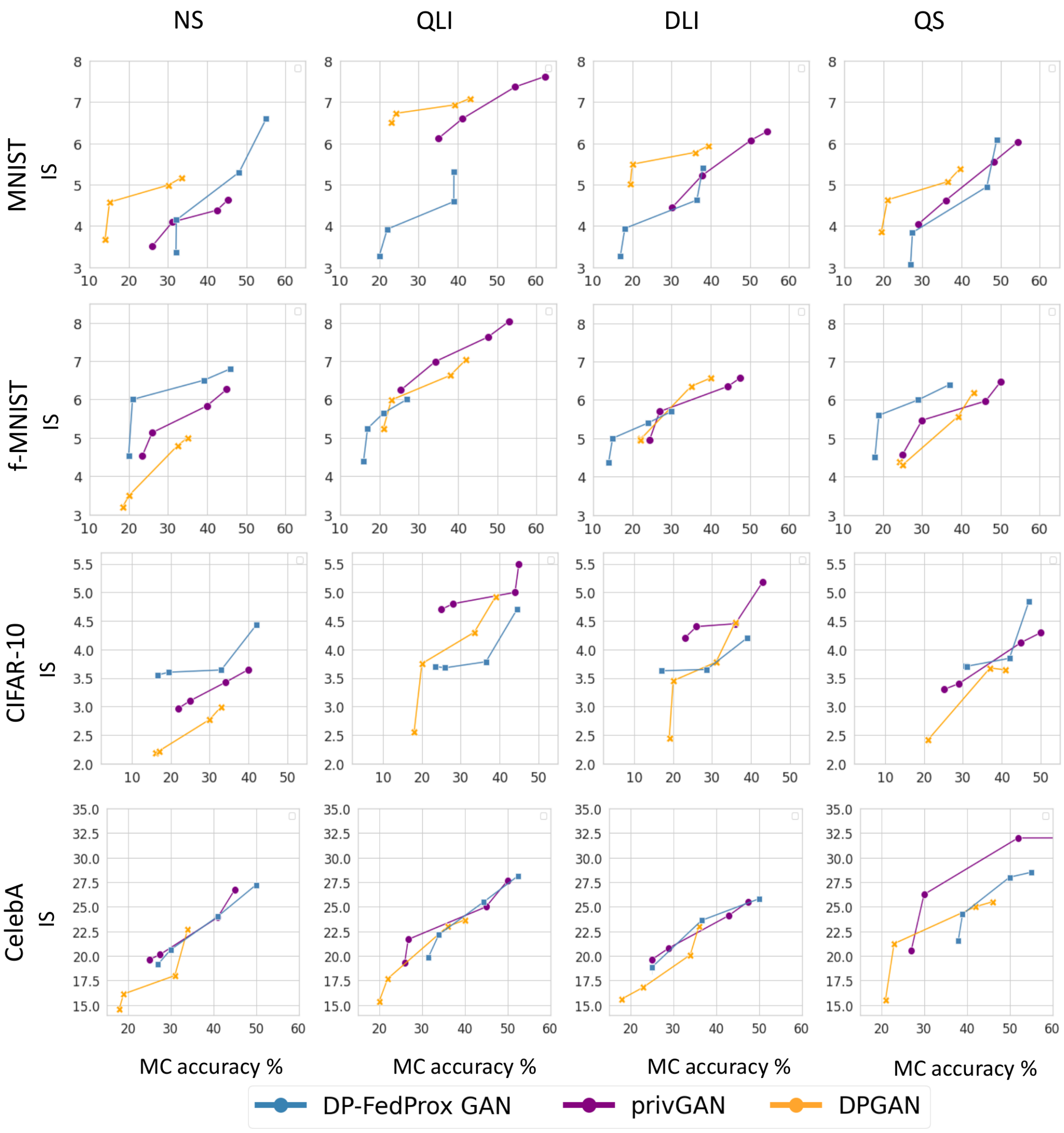


Fig. 20: Privacy vs. Utility: The Figure shows Monte-Carlo Attack vs Inception Score in privGAN, DPGAN, and DP-FedProx GAN for various privacy parameters.

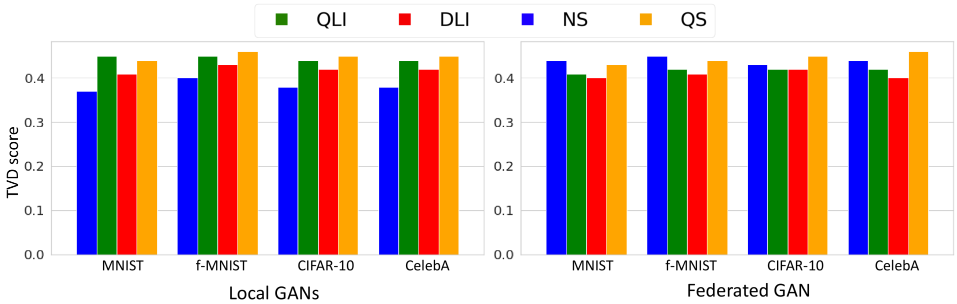


Fig. 21: TVD score for Non-Private Local GANs and Federated GAN: The Figure displays the results of a TVD attack score using non-private local GANs and federated GAN with $K = 10$ for various distributions for each dataset.

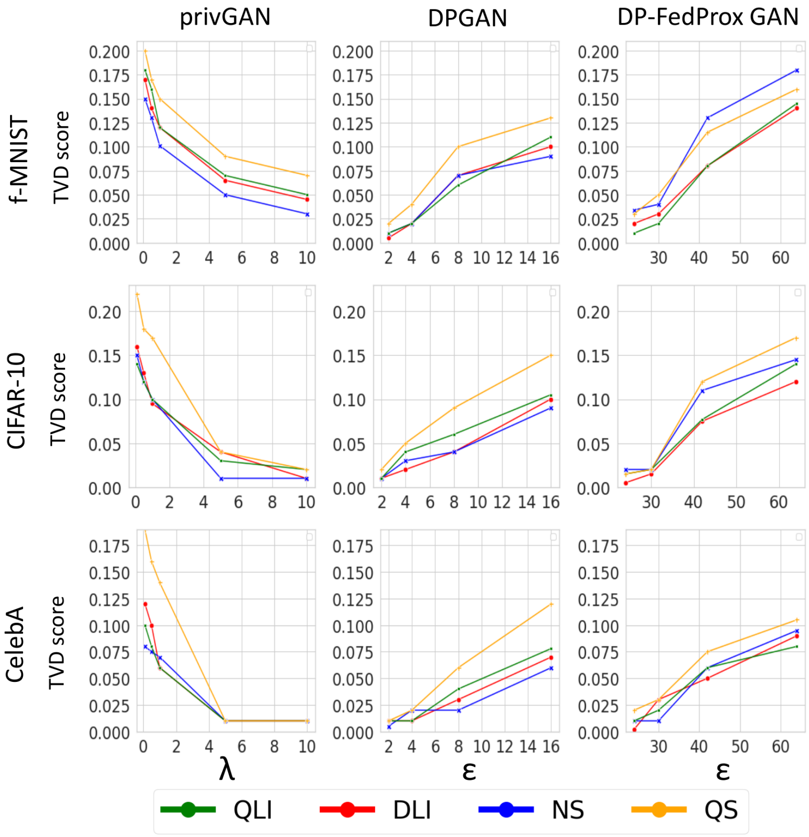


Fig. 22: Varying privacy parameters for TVD Attack: For different privacy parameters, the Figure displays the TVD attack score on the discriminator(s) in privGAN, DPGAN, and DP-FedProx GAN.

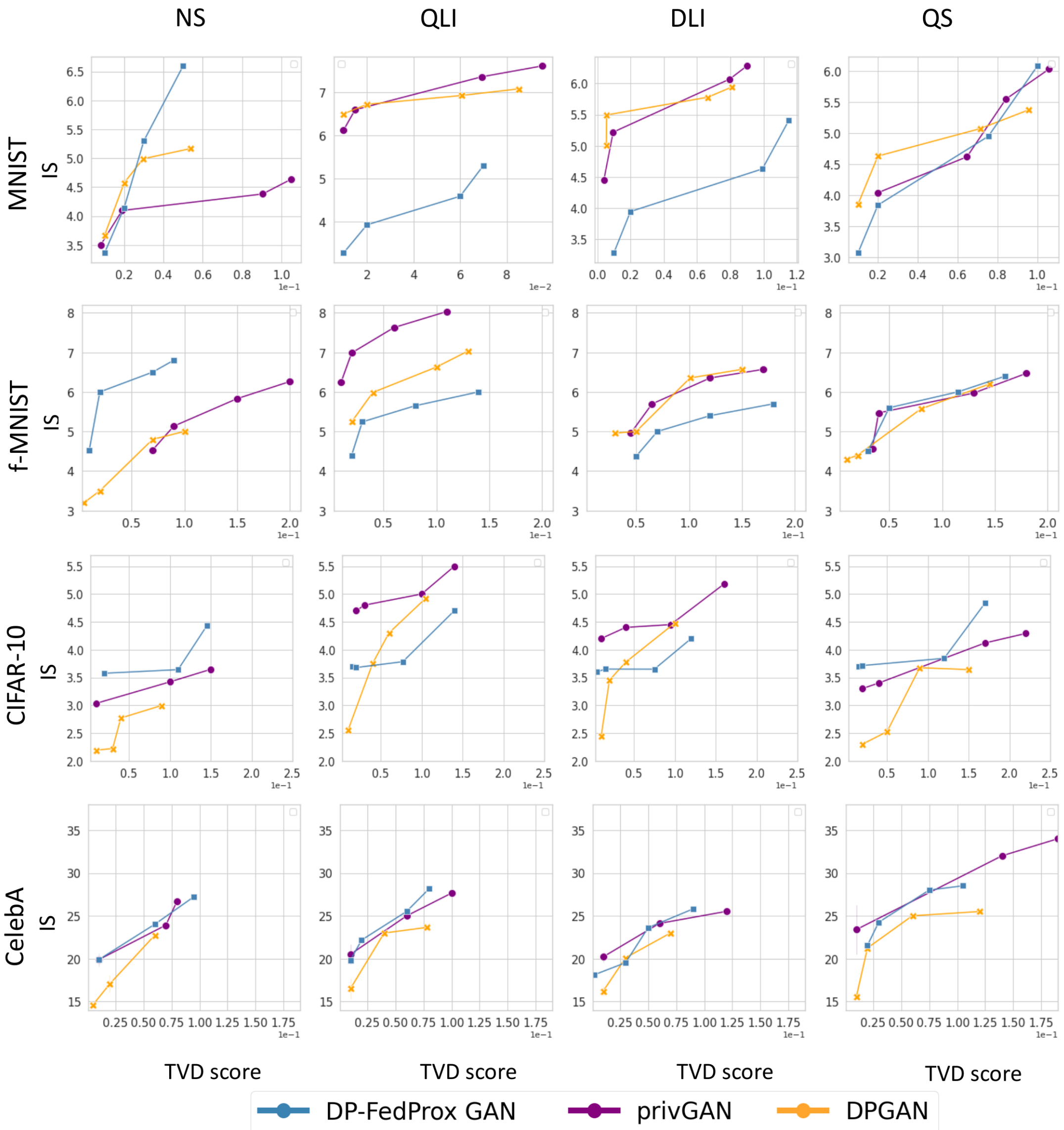


Fig. 23: Privacy vs. Utility: The Figure shows TVD Attack vs Inception Score in privGAN, DPGAN, and DP-FedProx GAN for various privacy parameters.