PRIVATE GANS, REVISITED

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that with improved training, the standard approach for differentially private GANs – updating the discriminator with noisy gradients – achieves or competes with state-of-the-art results for private image synthesis. Existing instantiations of this approach neglect to consider how adding noise only to discriminator updates disrupts the careful balance between generator and discriminator necessary for successful GAN training. We show that a simple fix restores parity: taking more discriminator steps between generator steps. Finally, with the goal of restoring parity between generator and discriminator, we experiment with further modifications to improve discriminator training and see further improvements. For MNIST at $\varepsilon = 10$, our private GANs improve the record FID from 48.4 to 13.0, as well as downstream classifier accuracy from 83.2% to 95.0%.

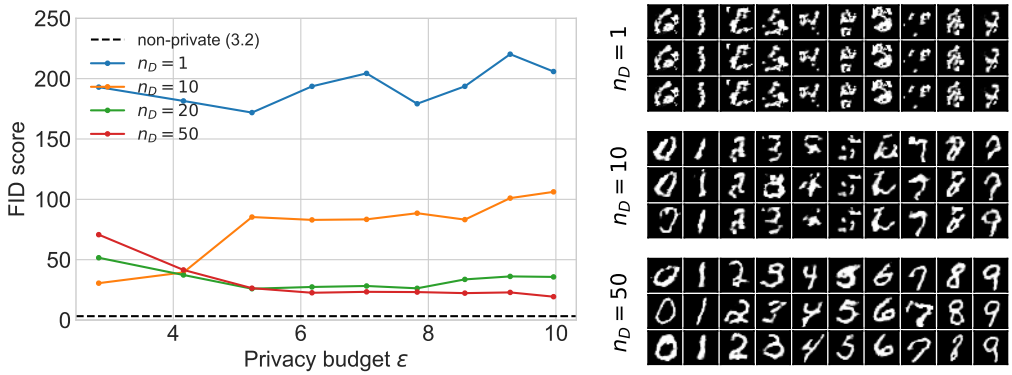
1 INTRODUCTION

Differential privacy (DP) (Dwork et al., 2006b) has emerged as a compelling approach for training machine learning (ML) models on sensitive data. However, incorporating DP requires significant changes to the model training process. Notably, it prevents the modeller from working directly with private data, complicating debugging and exploration. Furthermore, a data analyst can no longer interact with a private dataset after exhausting their allocated private budget. One approach to alleviate these issues is by producing *private synthetic data*, which can be plugged directly into existing ML pipelines, without further concern for privacy.

A line of work has studied leveraging deep generative models to produce DP synthetic data. Initial efforts focused on privatizing generative adversarial networks (GANs) (Goodfellow et al., 2014) by using differentially private stochastic gradient descent (DPSGD) (Abadi et al., 2016) to update the GAN discriminator – an approach referred to as *DPGAN* (Xie et al., 2018; Beaulieu-Jones et al., 2019; Torkzadehmahani et al., 2019). Follow-up work has studied alternative approaches to privatizing GANs (Jordon et al., 2019; Long et al., 2021; Chen et al., 2020; Wang et al., 2021), as well as other generative modelling frameworks, such as maximum mean discrepancy (Harder et al., 2021; Vinaroz et al., 2022), Sinkhorn divergences (Cao et al., 2021), and energy-based models (Chen et al., 2022).

Towards generating high-dimensional, complex data, these studies have primarily focused on image synthesis, which has served as the testbed of choice for research in (non-private) generative modelling. For the task of labelled image synthesis, the literature has corroborated that departing from the baseline DPGAN approach, either in the privatization scheme, or modelling framework altogether, leads to significant improvements in generation quality. Proposed explanations attribute these results to inherent limitations of the DPGAN framework, suggesting that: (1) privatizing discriminator training is sufficient for privacy, but may be overkill when only the generator needs to be released (Long et al., 2021); or (2) adversarial objectives may be unsuited for training under privacy (Cao et al., 2021).

We demonstrate that the reported poor results of DPGANs should not be attributed to inherent limitations of the framework, but rather, training issues. More precisely: we propose that the *asymmetric noise addition* in DPGANs (adding noise to discriminator updates only) weakens the discriminator relative to the generator, disrupting the careful balance necessary for successful GAN training. To account for this, we propose to take more discriminator steps between generator updates. With this change, DPGANs improve significantly (see Figure 1), achieving or competing with state-of-the-art results in private image synthesis.



(a) MNIST FID over a training run

(b) Corresponding images at $(10, 10^{-5})$-DP

Figure 1: Our DPGAN results on labelled MNIST synthesis at $(10, 10^{-5})$-DP. **Left:** We find that increasing n_D, the number of discriminator steps taken between generator steps, significantly improves image synthesis results. Using $n_D = 50$ instead of $n_D = 1$ on our baseline DPGAN improves FID from $205.9 \rightarrow 19.4$, which is also an improvement over the record FID of 48.4 from Cao et al. (2021). $n_D = 50$ also improves downstream classification accuracy to 92.9% (see Figure 2a), improving over the the record accuracy of 83.2% from Cao et al. (2021). **Right:** Corresponding synthesized images (please zoom in). We observe that large n_D improves visual quality, and low n_D leads to mode collapse.

Furthermore, we demonstrate that this perspective on private GAN training ("restoring parity to a discriminator weakened by DP noise") is an effective heuristic for improving private GANs. We make other modifications to discriminator training – large batch sizes and discriminator step schedules – to further improve upon the aforementioned results.

In summary, we make the following contributions:

- We diagnose the issue of *asymmetric noise addition* as a key factor in the degradation of performance observed when privatizing GANs. We show that taking more discriminator steps between generator updates (to account for noise) improves DPGANs.
- Based on this perspective, we show following the heuristic of improving classifiers allows us to design better private GAN training recipes that further improve over the aforementioned results.

2 Preliminaries

In this section, we give some background on differential privacy and present the standard approach for training GANs with differential privacy.

2.1 Differential Privacy

Our goal is to train a generative model on sensitive data that is safe to release, i.e., it does not leak the secrets of individuals in the training dataset. We do this by ensuring the training algorithm $\mathcal{A}$ – which takes as input the sensitive dataset $D \in \mathcal{U}$ and returns the parameters of a trained (generative) model $\theta \in \Theta$ – satisfies differential privacy.

Definition 1 (Differential Privacy (Dwork et al., 2006b))**.** A randomized algorithm $\mathcal{A} : \mathcal{U} \rightarrow \Theta$ is (ε, δ)-*differentially private* if for every pair of neighbouring datasets $D, D' \in \mathcal{U}$, we have

$$\mathbb{P}[\mathcal{A}(D) \in S] \leq \exp(\varepsilon) \cdot \mathbb{P}[\mathcal{A}(D') \in S] + \delta \quad \text{for all } S \subseteq \Theta.$$

In this work, we adopt the add/remove definition of DP, and say two datasets D and D' are neighbouring if they differ in at most one entry, that is, $D = D' \cup \{x\}$ or $D' = D \cup \{x\}$.

A gradient-based training algorithm can be privatized by employing *differentially private stochastic gradient descent* (DPSGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016) as a drop-in

replacement for SGD. DPSGD involves clipping per-example gradients and adding Gaussian noise to their sum, which effectively bounds and masks the contribution of any individual point to the final model parameters.

Privacy analysis of DPSGD follows from a number of classic tools in the DP toolbox: the Gaussian mechanism, privacy amplification by subsampling, and composition (Dwork et al., 2006a; Dwork & Roth, 2014; Balle et al., 2018; Wang et al., 2019). More recently, *exact* privacy accounting has been enabled by a line of work focused on composition via discretization and the fast Fourier transform applied to the privacy loss random variable (Koskela et al., 2020; Gopi et al., 2021; Zhu et al., 2022). In this work, we employ the DPSGD analysis of Mironov et al. (2019) implemented in Opacus (Yousefpour et al., 2021).

We highlight one convenient property of DP, known as *closure under post-processing*. This says that interacting with a privatized model (e.g., using it to compute gradients on non-sensitive data, drawing samples from it) does not lead to any further privacy violation.

Theorem 2 (Post-processing)**.** Let $\mathcal{A} : \mathcal{U} \to \Theta$ be a randomized algorithm that is (ε, δ)-DP, and $f : \Theta \to \mathcal{Y}$ be an arbitrarily randomized mapping. Then $f \circ \mathcal{A} : \mathcal{U} \to \mathcal{Y}$ is (ε, δ)-DP.

2.2 Differentially Private GANs

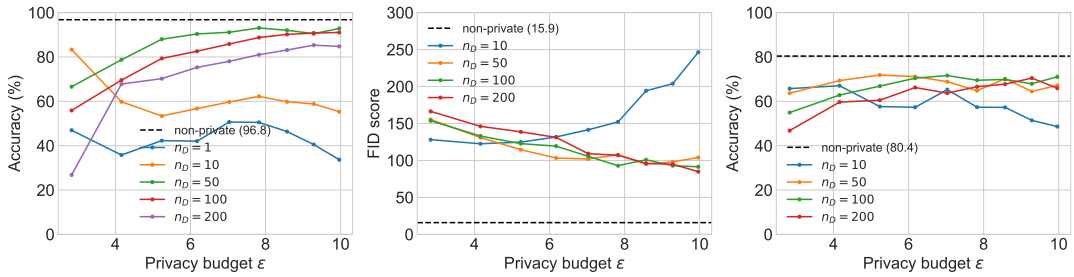
Algorithm 1 details the training algorithm for DPGANs, which is effectively an instantiation of DPSGD. Note that only gradients for the discriminator $\mathcal{D}$ must be privatized (via clipping and noise), and not those for the generator $\mathcal{G}$. This is because the generator only interacts with the sensitive dataset through the discriminator, and thus does not need further privatization due to post-processing (Theorem 2).

Algorithm 1 TrainDPGAN$(D; \cdot)$

Input: Labelled dataset $D = \{(x_j, y_j)\}_{j=1}^n$. Discriminator $\mathcal{D}$ and generator $\mathcal{G}$ initializations ϕ_0 and θ_0. Hyperparameters n_D, T, B, C, σ. Optimizers `OptD`, `OptG`. Privacy parameter δ.
$q \leftarrow B/|D|$ and $t, k \leftarrow 0$ ▷ calculate sampling rate q, initialize counters
while $t < T$ **do** ▷ update $\mathcal{D}$ with DPSGD
  $S_t \sim$ PoissonSample(D, q) ▷ sample a real batch S_t by including each $(x, y) \in D$ w.p. q
  $\widetilde{S}_t \sim \mathcal{G}(\cdot; \theta_k)^B$ ▷ sample fake batch $\widetilde{S}_t$
  $g_{\phi_t} \leftarrow \sum_{(x,y)\in S_t} \text{clip}\left(\nabla_{\phi_t}(-\log(\mathcal{D}(x, y; \phi_t))); C\right)$
    $+ \sum_{(\widetilde{x},\widetilde{y})\in \widetilde{S}_t} \text{clip}\left(\nabla_{\phi_t}(-\log(1 - \mathcal{D}(\widetilde{x}, \widetilde{y}; \phi_t))); C\right)$ ▷ clipped per-example gradients
  $\widehat{g}_{\phi_t} \leftarrow \frac{1}{2B}(g_{\phi_t} + z_t)$, where $z_t \sim \mathcal{N}(0, C^2\sigma^2 I))$ ▷ add Gaussian noise
  $\phi_{t+1} \leftarrow$ `OptD`$(\phi_t, \widehat{g}_{\theta_t})$ and $t \leftarrow t + 1$
  if n_D divides t **then** ▷ perform $\mathcal{G}$ update every n_D steps
    $\widetilde{S}'_t \sim \mathcal{G}(\cdot; \theta_k)^B$
    $g_{\theta_k} \leftarrow \frac{1}{B} \sum_{(\widetilde{x},\widetilde{y})\in \widetilde{S}'_t} \nabla_{\theta_k}(-\log(\mathcal{D}(\widetilde{x}, \widetilde{y}; \phi_t)))$
    $\theta_{k+1} \leftarrow$ `OptG`(θ_k, g_{θ_k}) and $k \leftarrow k + 1$
  end if
end while
$\varepsilon \leftarrow$ PrivacyAccountant(T, σ, q, δ) ▷ compute privacy budget spent
Output: Final $\mathcal{G}$ parameters θ_k. (ε, δ)-DP guarantee.

3 Frequent discriminator steps improves private GANs

In this section, we discuss our main finding: the number of discriminator steps taken between each generator step (n_D from Algorithm 1) plays a significant role in the success of private GAN training. For a fixed setting of DPSGD hyperparameters, there is an optimal range of values for n_D that maximizes generation quality, in terms of both visual quality and utility for downstream classifier training. This value is often quite large ($n_D \approx 100$ in some cases).



(a) MNIST accuracy (b) FashionMNIST FID (c) FashionMNIST accuracy

Figure 2: DPGAN results targeting $(10, 10^{-5})$-DP under different discriminator update frequencies n_D. **(a)** As a measure of synthetic data utility, we plot the test set accuracy of a CNN trained on generated data only. Accuracy mirrors FID scores (see Figure 1). Going from to $n_D = 1$ to $n_D = 50$ improves accuracy from $33.7\% \to 92.9\%$. Further increases hurts accuracy. **(b) & (c)** We obtain similar results for FashionMNIST. Note that the optimal n_D is higher (around $n_D \approx 100$). At $n_D = 100$, we obtain an FID of 91.5 and accuracy of 71.1%, which compares favourably to the record FID of 128.3 and record accuracy of 75.5% reported in Cao et al. (2021) for $(10, 10^{-5})$-DP generation of FashionMNIST.

3.1 Baseline experimental setup

We focus on labelled generation of MNIST (LeCun et al., 1998) and FashionMNIST (Xiao et al., 2017), both of which are comprised of $60,000$ 28×28 grayscale images divided into 10 classes. To build a strong baseline, we begin from an open source PyTorch (Paszke et al., 2019) implementation[1] of DCGAN (Radford et al., 2016) that performs well non-privately, and copy their training recipe. This includes: batch size $B = 128$, the Adam optimizer (Kingma & Ba, 2015) with parameters $(\alpha = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999)$ for both $\mathcal{G}$ and $\mathcal{D}$, the non-saturating GAN loss (Goodfellow et al., 2014), and a 5-layer fully convolutional architecture with width parameter $d = 128$. Then, to adapt it to our purposes, we make three architectural modifications: in both $\mathcal{G}$ and $\mathcal{D}$ we (1) remove all BatchNorm layers (which are not compatible with DPSGD); (2) add label embedding layers to enable labelled generation; and (3) adjust convolutional/transpose convolutional stride lengths and kernel sizes as well as remove the last layer, in order to process $1 \times 28 \times 28$ images without having to resize. Finally, we remove their custom weight initialization, opting for PyTorch defaults. Training this configuration non-privately yields labelled generation that achieves FID scores of 3.2 on MNIST and 15.9 on FashionMNIST. Finally, we note that these models are not small: $\mathcal{D}$ and $\mathcal{G}$ have 1.72M and 2.27M trainable parameters respectively.

To privatize training, we use Opacus (Yousefpour et al., 2021) which implements per-example gradient computation and the RDP accounting of Mironov et al. (2019). For our baseline setting, we use the following DPSGD hyperparameters: we keep the (expected) batch size $B = 128$, and use a noise scale $\sigma = 1$ and clipping norm $C = 1$. Under these settings, we have the budget for $T = 450,000$ discriminator steps when targeting $(10, 10^{-5})$-DP.

When training the discriminator, we draw B fake examples and compute clipped per-example gradients on the entire combined batch of real and fake examples (see Algorithm 1). This is the approach taken in the prior work of Torkzadehmahani et al. (2019). We remark that this is purely a design choice: it is not necessary to clip the gradients of the fake samples, or process them together in the same batch. So long as we preserve the sensitivity of gradient queries *with respect to the real data*, the same amount of noise will suffice for privacy.

3.2 Evaluation

We evaluate our generative models by examining the *visual quality* and *utility for downstream tasks* of generated images. Following prior work, we measure visual quality by computing the Fréchet Inception Distance (FID) (Heusel et al., 2017) between $60,000$ generated images and $10,000$ real

[1] Courtesy of Hyeonwoo Kang (`https://github.com/znxlwm`). Code available at this link.

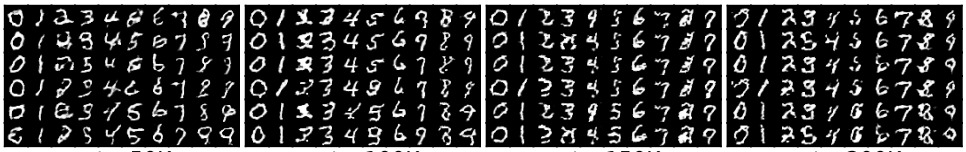


Figure 3: Evolution of samples drawn during training at $n_D = 10$, when targeting $(10, 10^{-5})$-DP. This setting reports its best FID and downstream accuracy at $t = 50$K iterations ($\varepsilon \approx 2.85$). As training progresses, we observe mode collapse for several classes alongside the deterioration in evaluation metrics.

images from the test set.[2] FID compares statistics between deep feature representations of image datasets, and has been shown to be correlated with human judgments of visual quality. To measure downstream task utility, we again follow prior work, and train a CNN classifier on $60,000$ generated image-label pairs and report its accuracy on the real test set.

3.3 More frequent discriminator steps improves generation quality

We plot in Figures 1 and 2 the evolution of FID and downstream accuracy during DPGAN training for both MNIST and FashionMNIST, under varying discriminator update frequencies n_D. The effect of this parameter has outsized impact on the final results. For MNIST, $n_D = 50$ yields the best results; on FashionMNIST, the best FID is obtained at $n_D = 200$ and the best accuracy at $n_D = 100$.

We emphasize that varying the *frequency* of discriminator steps, relative to generator steps, does not affect the privacy cost of Algorithm 1. For any setting of n_D, we perform the same number of noisy gradient queries on real data – what changes is the total number of generator steps taken over the course of training, which is reduced by a factor of $1/n_D$.

3.4 Private GANs are on a path to mode collapse

In Figures 1 and 2a, we observe that at low discriminator update frequencies ($n_D = 10$), the best FID and accuracy scores occur early in training, well before the privacy budget we are targeting is exhausted.[3] At $50,000$ discriminator steps ($\varepsilon \approx 2.85$), $n_D = 10$ has better FID (30.6) and accuracy (83.3%) than the other settings. However, these results degrade by the end of training. In Figure 3, we plot the evolution of generated images in this setting over the course of training, and observe qualitative evidence of mode collapse, co-occurring with deterioration in evaluation metrics.

These results suggest that **fixing other DPSGD hyperparameters, there is an optimal setting for the discriminator step frequency** n_D that strikes a balance between: (1) being too low, causing the model performance peak early in training and undergo mode collapse; resulting in all subsequent training to consume additional privacy budget *without improving the model*; and (2) being too high, preventing the generator from taking enough steps to converge before the privacy budget is exhausted (an example of which is pictured in Figure 2a). Striking this balance results in the most effective utilization of privacy budget towards improving the generator.

4 Why does taking more steps help?

In this section, we present empirical findings towards understanding why more frequent discriminator steps improves DPGAN training.

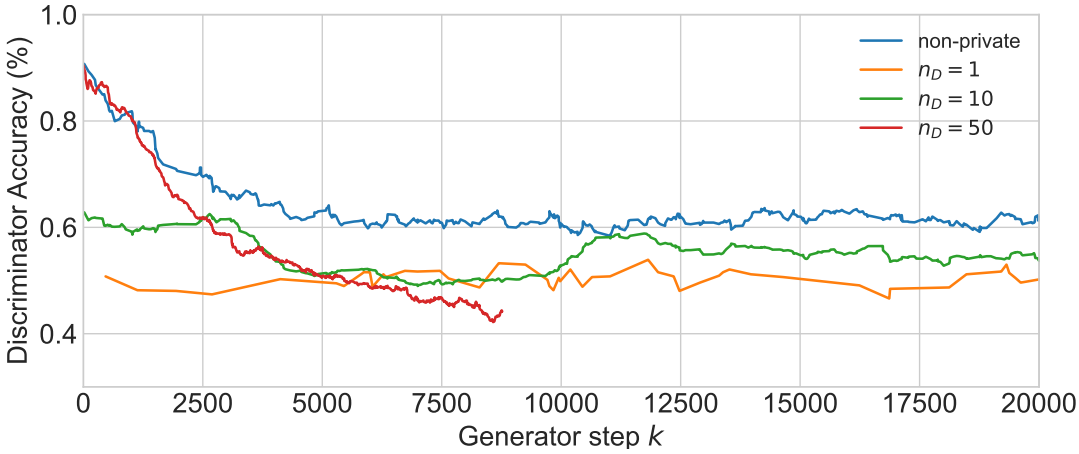


Figure 4: Discriminator accuracy immediately before each generator step. While non-privately the discriminator maintains a 60% accuracy, the private discriminator with $n_D = 1$ is effectively a random guess. Increasing the number of discriminator steps recovers the discriminator's advantage early in the process.

4.1 How does DP affect GAN training?

Figure 4 compares the accuracy of the GAN discriminator (on held-out real and fake examples) immediately before each generator step, between non-private training and simple privatization ($n_D = 1$). We observe that non-privately, discriminator accuracy stays around 60% throughout training. However, introducing DP leads to a qualitative difference: DP causes discriminator accuracy to drop to 50% immediately at the start of training, to never recover.[4]

We also plot discriminator accuracy immediately before a generator update for various settings of n_D. We make three observations: (1) Larger n_D corresponds to higher accuracy; (2) The generator improves during the periods in which the discriminator stays above 50% accuracy; and (3) accuracy decreases throughout training, and the degradation of results co-occurs with accuracy of the discriminator dropping below 50%.

Based on these observations, we propose the following explanation:

- Generator improvement occurs when our discriminator is capable of distinguishing between real and fake data.
- The asymmetric noise addition introduced by DP to the discriminator makes the task difficult.
- Allowing the discriminator to train longer on a fixed generator improves its accuracy, recovering the non-private case where the generator and discriminator are balanced.

4.2 Does reducing noise accomplish the same thing?

In light of this explanation, we ask if reducing the noise level can offer the same improvement as taking more steps. To test this: starting from our setting in Section 3, fixing $n_D = 1$, and targeting MNIST at $\varepsilon = 10$, we search over a grid of noise levels $\sigma = \{0.4, 0.43, 0.45, 0.5, 0.55, 0.6, 0.7, 0.8\}$; the lowest of which, $\sigma = 0.4$, admits a budget of only

[2]We use an open source PyTorch implementation to compute FID: https://github.com/mseitzer/pytorch-fid

[3]This observation has been reported in (Neunhoeffer et al., 2021), serving as motivation for their remedy of taking a mixture of intermediate models encountered in training. We are not aware of any mentions of this aspect of DPGAN training in papers reporting DPGAN baselines for labelled image synthesis.

[4]Our plot only shows the first $20,000$ generator steps, but we remark that this persists until the end of training ($450,000$ steps).

$T = 360$ discriminator steps. We obtain a best FID of 127.1 and best accuracy of 57.5% at noise level $\sigma = 0.45$. Hence we can conclude that in this setting, incorporating discriminator update frequency in our design space leads to better models.

As we discuss in more detail in Section 5.1, when we are able to find other means to improve the discriminator beyond taking more steps, tuning discriminator update frequency may not yield improvements. To illustrate with an extreme case, consider eliminating the privacy constraint. In non-private GAN training, taking more steps is known to be unnecessary, potentially hurting results.

5 Better generators via better discriminators

Our proposed explanation in Section 4 provides a concrete suggestion for improving GAN training: effective use of our privacy budget to maximize the number of generator steps taken when the discriminator has a high accuracy will improve private generation. We experiment with modifications to the private GAN training recipe to improve discriminator accuracy, which translate to improved generation.

5.1 Larger batch sizes

Several recent works have demonstrated that for classification tasks, large batch sizes improve the performance of DPSGD, after tuning for the optimal noise scale σ accordingly (Tramèr & Boneh, 2021; Anil et al., 2021; De et al., 2022). GAN training is typically conducted with small batch sizes (for example, DCGAN uses $B = 128$, which we adopt; StyleGAN uses $B = 32$). Therefore it is interesting to see if large batch sizes indeed improve private GAN training.

We scale up batch sizes, considering $B \in \{64, 128, 512, 2048\}$, and search for the optimal noise scale σ and n_D. We target two settings, the high privacy ($\varepsilon = 1$) and low privacy ($\varepsilon = 10$) regimes. For batch size 128 targeting $\varepsilon = 10$, we search over three noise scales, $\Sigma_{128} = \{0.6, 1.0, 1.4\}$. We choose candidate noise scales for other batch sizes as follows: when considering a batch size $128k$, we search over $\Sigma_{128k} = \{\sqrt{k} \cdot \sigma : \sigma \in \Sigma_{128}\}$. For $\varepsilon = 1$, we multiply all noise scales by 5. We search over a grid $n_D \in \{1, 2, 5, 10, 20, 50, 100, 200, 500\}$, omitting some values that we are confident will fail due to compute limitations (e.g., trying $n_D = 1$ when mode collapse occurs for $n_D = 5$).

We report the best results from our hyperparameter search in in Table 1. We find that larger batch sizes leads to improvements: for $\varepsilon = 10$, the best MNIST and FashionMNIST results are achieved at $B = 2048$. For $\varepsilon = 1$, the best results are achieved at $B = 512$. We also note that for large batch sizes, the optimal number of generator steps can be quite small. For $B = 2048$, $\sigma = 4.0$, targeting MNIST at $\varepsilon = 10$, $n_D = 5$ is the optimal discriminator update frequency, and improves over our best $B = 128$ setting employing $n_D = 50$.

5.2 Discriminator step scheduling

Our observations from Section 3 and 4 motivate us to consider a discriminator step frequency schedule. As pictured in Figure 4, discriminator accuracy drops during training as the quality of generated images improve. We simultaneously observe degraded performance. Therefore, we would like the ability to take more steps at that point to improve the discriminator, before taking a generator update. However, using a large discriminator update frequency at the beginning of training is wasteful – we end up using a a lot our privacy budget on early steps, which we can afford to skimp on, as evidenced by the fact that low n_D achieves the best FID and accuracy early in training. Hence we propose to start at a low discriminator update frequency, and ramp up when our discriminator is performing poorly.

The accuracy of our discriminator on a validation set is a function of the private data, and thus must be released with DP. While this is feasible, it introduces the additional problem of having to find the right split of privacy budget for the best performance. We observe that discriminator accuracy is related to discriminator accuracy on only generated samples (which are free to evaluate on, by post-processing). Hence we use it as a proxy to assess discriminator performance.

The step schedule is parameterized by two terms, β and d. β is the decay parameter used to compute the exponential moving average (EMA) of discriminator accuracy on fake batches before each generator update. We use $\beta = 0.99$ in all settings. d is the accuracy threshold that upon falling below, we increase the update frequency. We try $d = 0.6$ and $d = 0.7$, finding that 0.7 works better for large batches. Additionally, we promise a grace period of $2/(1-\beta) = 200$ generator steps before moving on to the next update frequency. This formula is motivated by the fact that β-EMA's value is primarily determined by its last $2/(1-\beta)$ observations.

The additional benefit of the step schedule is that it means we do not have to search for the optimal update frequency. Although the step schedule introduces the extra hyperparameter of the threshold d, we found that these two settings ($d = 0.6$ and 0.7) were sufficient to improve over settings that were the result of a much more extensive hyperparameter search.

5.3 Comparison with state of the art

Privacy Level	Method	Reported In	MNIST		FashionMNIST	
			FID	Acc.(%)	FID	Acc.(%)
$\varepsilon = \infty$	Real data	(This work)	1.0	99.2	1.5	92.5
	GAN		3.2	96.8	15.9	80.4
$\varepsilon = 10$	DPGAN	Chen et al. (2020)	179.16	63	243.80	50
		Long et al. (2021)	304.86	80.11	433.38	60.98
	GS-WGAN	Chen et al. (2020)	61.34	80	131.34	65
	PATE-GAN	Long et al. (2021)	253.55	66.67	229.25	62.18
	G-PATE	Long et al. (2021)	150.62	80.92	171.90	69.34
	Datalens	Wang et al. (2021)	173.50	80.66	167.68	70.61
	DP-MERF	Cao et al. (2021)	116.3	82.1	132.6	**75.5**
	DP-Sinkhorn	Cao et al. (2021)	48.4	83.2	128.3	75.1
	DPGAN	(This work)	19.4	92.9	91.5	71.1
	+ large batches		13.2	94.3	66.7	72.1
	+ step schedule		**13.0**	**95.0**	**56.8**	74.8
$\varepsilon = 1$	DPGAN	Long et al. (2021)	470.20	40.36	472.03	10.53
	GS-WGAN	Long et al. (2021)	489.75	14.32	587.31	16.61
	PATE-GAN	Long et al. (2021)	231.54	41.68	253.19	42.22
	G-PATE	Long et al. (2021)	153.38	58.80	214.78	58.12
	Datalens	Wang et al. (2021)	186.06	71.23	194.98	64.78
	DP-MERF	Jiang et al. (2022)	118.3	80.5	**102.1**	**74.6**
	DP-MERF	Vinaroz et al. (2022)	-	80.7	-	73.9
	DP-HP	Vinaroz et al. (2022)	-	**81.5**	-	72.3
	DPGAN	(This work)	91.7	77.4	151.9	65.0
	+ large batches		66.1	73.7	153.2	66.6
	+ step schedule		**56.2**	80.1	121.8	68.0

Table 1: We gather previously reported results in the literature on the performance of various methods for labelled generation of MNIST and FashionMNIST. Note that *Reported In* refers to the source of the numerical result, not the originator of the approach. For downstream accuracy, we report the best accuracy among classifiers they use, and compare against our CNN classifier accuracy.

Table 1 summarizes our best experimental settings. In the low privacy/high ε regime, most of our results are dramatically better than prior work[5] – for example, decreasing FID from 48.4 to 13.0 and increasing accuracy from 83.2% to 95.0% on MNIST. In the high privacy/low ε regime, im-

[5]We do not compare with two recent works on private generative models (Chen et al., 2022; Jiang et al., 2022), as we believe there are gaps in their privacy analyses. This has been confirmed by the authors of Jiang et al. (2022), and the sketch of an argument regarding non-privacy of Chen et al. (2022) has been shared with us by others (Anonymous, 2022).

provements are not quite as extreme, but can still be significant (FID for MNIST), and only compare negatively to state-of-the-art for accuracy on Fashion MNIST. Broadly speaking, we see an improvement from taking larger batch sizes, and then another with an adaptive step schedule. We provide some example generated images in Figures 5 and 6.

6 Discussion and Related work

DP generative modelling. Differentially private GANs (DPGANs) were introduced by Xie et al. (2018). While a number of subsequent works studied and tuned these models in a variety of settings (Torkzadehmahani et al., 2019; Jordon et al., 2019; Long et al., 2021; Chen et al., 2020; Wang et al., 2021), none of them considered a significantly larger number of discriminator steps per generator step. Other private generative models include VAEs (Chen et al., 2018; Takahashi et al., 2020), mixture models (Acs et al., 2018), maximum mean discrepancy (Harder et al., 2021), Sinkhorn divergences (Cao et al., 2021), energy-based models (Chen et al., 2022), and normalizing flows (Waites & Cummings, 2021). We show that a well-tuned DPGAN competes or outperforms these approaches.

Custom approaches versus a well-tuned DPSGD. An ongoing debate pertains to the best techniques and architectures for private ML. Roughly speaking, there are two schools of thought. One investigates novel architectures for privacy, which may be outperformed by more traditional approaches in the non-private setting. Some examples include Chen et al. (2018); Cao et al. (2021); Vinaroz et al. (2022), a variety of generative models specifically designed to be compatible with differential privacy. The other focuses on searching within the space of tried-and-tested methods that are understood to work well non-privately. Some examples include the works of De et al. (2022); Li et al. (2022); Yu et al. (2022), who demonstrate that, similar to the non-private setting, large-scale CNN and Transformer architectures can achieve state-of-the-art results for image classification and NLP tasks. The primary modifications to the pipeline are along the lines of changing the batch size, modifying the type of normalization layers, etc., most of which would be explored in a proper hyperparameter search in the non-private setting. Our work fits into the latter line: we show that novel generative models introduced for privacy can be outperformed by GANs trained with well-tuned DPSGD.

Tabular data. Our investigation focused on image datasets, namely MNIST and Fashion MNIST, while many important applications of private data generation involve *tabular data*. While Tao et al. (2021) find that private GAN-based approaches fail to preserve even basic statistics in these settings, we believe that our techniques may yield similar improvements.

Guidance in the non-private setting (tip 14 of Chintala et al. (2016)) prescribes to train the discriminator for more steps, especially in the presence of noise (a regularization approach used in non-private GANs). This is the case for DP, and is essentially our core strategy that yields the most significant gains in utility. We were not aware of this tip when we discovered this phenomenon, but it serves as validation of our finding. While Chintala et al. (2016) provide little elaboration, looking at further explorations of this principle in the non-private setting may provide guidance for improving private GANs.

Our most important contribution is to show that private GANs have been underrated by the research community, and in fact can achieve state-of-the-art results by properly exploring the search space. We hope and anticipate this will inspire the community to revisit private GANs, and quickly improve upon our results.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. ACM.

[5]We group per-class unconditional GANs together with conditional GANs under the DPGAN umbrella.

[5]These results are presented graphically in the paper. Exact numbers can be found in their code.

Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1109–1121, 2018.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.

Anonymous. Personal communication, 2022. Anonymous for double blind.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pp. 6277–6287. Curran Associates, Inc., 2018.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pp. 464–473, Washington, DC, USA, 2014. IEEE Computer Society.

Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P. Bhavnani, James Brian Byrd, and Casey S. Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7), 2019.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, 2021.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, 2020.

Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. DPGEN: differentially private generative energy-guided network for natural image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pp. 8387–8396, 2022.

Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*, 2018.

Soumith Chintala, Emily Denton, Martin Arjovsky, and Michael Mathieu. How to train a gan? tips and tricks to make gans work. *GitHub, Dec*, 2016.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pp. 486–503, Berlin, Heidelberg, 2006a. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pp. 265–284, 2006b.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, 2014.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

Frederik Harder, Kamil Adamczewski, and Mijung Park. DP-MERF: differentially private mean embeddings with random features for practical privacy-preserving data generation. In *The 24th International Conference on Artificial Intelligence and Statistics (AISTATS'21)*, pp. 1819–1827, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30 (NIPS'17)*, 2017.

Dihong Jiang, Guojun Zhang, Mahdi Karami, Xi Chen, Yunfeng Shao, and Yaoliang Yu. DP^2-VAE: differentially private pre-trained variational autoencoders. *arXiv preprint arXiv:2208.03409*, 2022.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: generating synthetic data with differential privacy guarantees. In *7th International Conference on Learning Representations (ICLR'19)*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR'15)*, 2015.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, AISTATS '20, pp. 2560–2569. JMLR, Inc., 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-PATE: Scalable differentially private data generator via private aggregation of teacher discriminators. In *Advances in Neural Information Processing Systems 34 (NeurIPS '21)*, pp. 2965–2977, 2021.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *CoRR*, abs/1908.10530, 2019.

Marcel Neunhoeffer, Steven Wu, and Cynthia Dwork. Private post-gan boosting. In *ICLR*. OpenReview.net, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations (ICLR'16)*, 2016.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP '13, pp. 245–248, Washington, DC, USA, 2013. IEEE Computer Society.

Tsubasa Takahashi, Shun Takagi, Hajime Ono, and Tatsuya Komatsu. Differentially private variational autoencoders with term-wise gradient aggregation. *arXiv preprint arXiv:2006.11204*, 2020.

Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. *CoRR*, abs/2112.09238, 2021.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-CGAN: differentially private synthetic data and label generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, (CVPR Workshops '19)*, pp. 98–104, 2019.

Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *9th International Conference on Learning Representations (ICLR'21)*, 2021.

Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. Hermite polynomial features for private data generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pp. 22300–22324, 2022.

Chris Waites and Rachel Cummings. Differentially private normalizing flows for privacy-preserving density estimation. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 1000–1009, 2021.

Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. DataLens: scalable privacy preserving training via gradient compression and aggregation. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2146–2168, 2021.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, AISTATS '19, pp. 1226–1235. JMLR, Inc., 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Gosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in pytorch. *CoRR*, abs/2109.12298, 2021.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, AISTATS '22, pp. 4782–4817. JMLR, Inc., 2022.

A GENERATED SAMPLES

We provide a few non-cherrypicked samples from MNIST and Fashion MNIST.

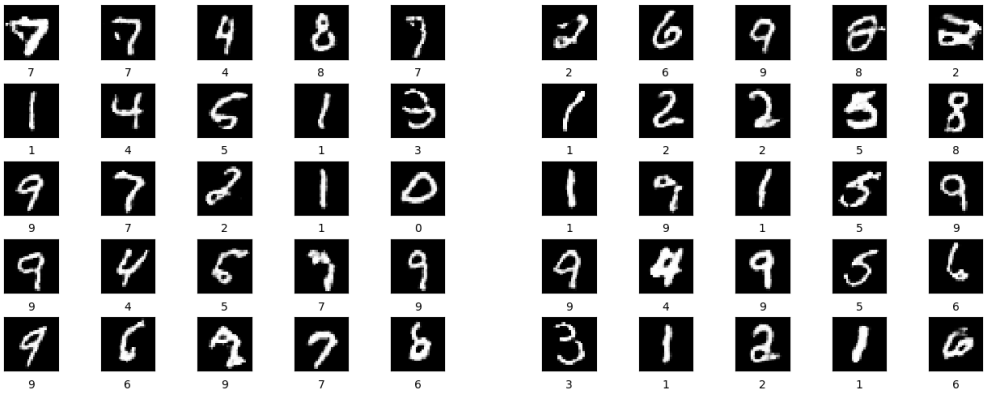


Figure 5: Some non-cherrypicked MNIST samples from our method, $\varepsilon = 10$.

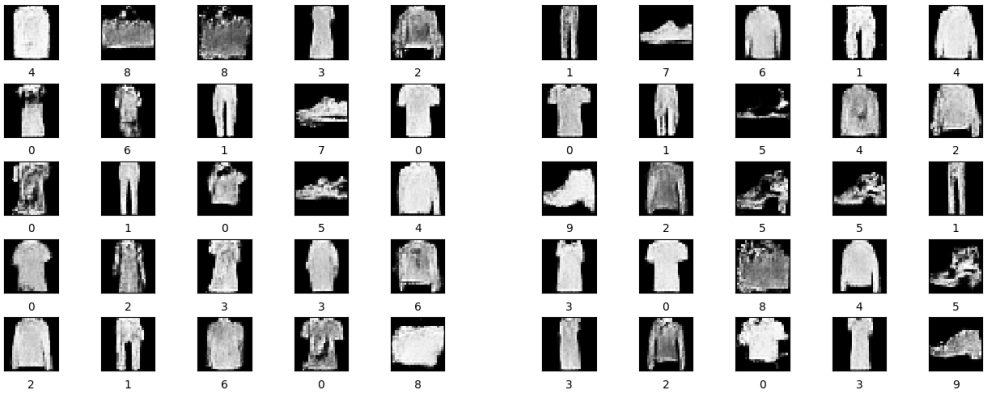


Figure 6: Some non-cherrypicked Fashion MNIST samples from our method, $\varepsilon = 10$.