Private GANs, Revisited

Anonymous Author(s)
Affiliation
Address
email

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 in generation quality. 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

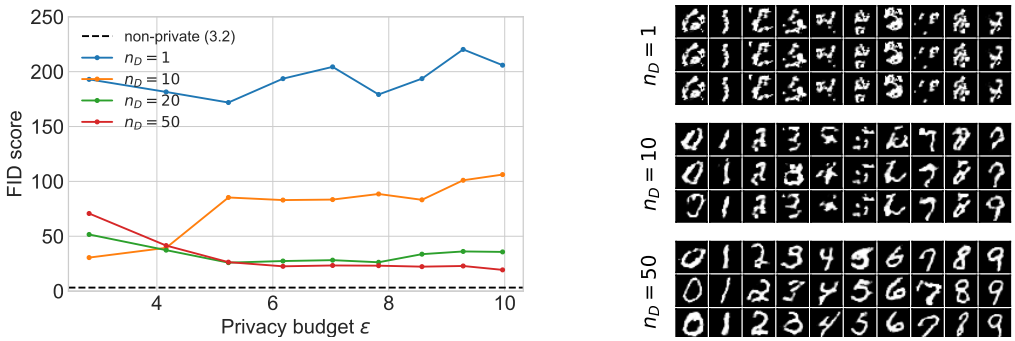
A recent line of work has studied leveraging deep generative models to produce DP synthetic data. Initial efforts focused on privatizing generative adversarial networks (GANs) [16] by using differentially private stochastic gradient descent (DPSGD) [1] to update the GAN discriminator – an approach referred to as *DPGAN* [35, 6, 29]. Follow-up work has studied alternative approaches to privatizing GANs [20, 22, 8, 33], as well as other generative modelling frameworks, such as maximum mean discrepancy [17, 31], Sinkhorn divergences [7], and energy-based models [9].

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 [22]; or (2) adversarial objectives may be unsuited for training under privacy [7].

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.

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 – and see futher improvements.

Submitted to NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research. Do not distribute.



(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 $205.9 \to 19.4$, which is also an improvement over the record FID of 48.4 from [7]. $n_D = 50$ also improves downstream classification accuracy to 92.9% (see Figure 2a), improving over the the record accuracy of 83.2% from [7]. **Right:** Corresponding synthesized images. We observe that large n_D improves visual quality, and low n_D leads to mode collapse.

2 Preliminaries

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 $D \in \mathcal{L}$ and returns the parameters of a trained (generative) model $\theta \in \Theta$ – satisfies differential privacy.

Definition 1 (Differential Privacy [15])**.** A randomized algorithm $\mathcal{A} : \mathcal{L} \to \Theta$ is (ε, δ)-*differentially private* if for every pair of neighbouring datasets $D, D' \in \mathcal{L}$, we have

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

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 $D' = D \cup \{x\}$.

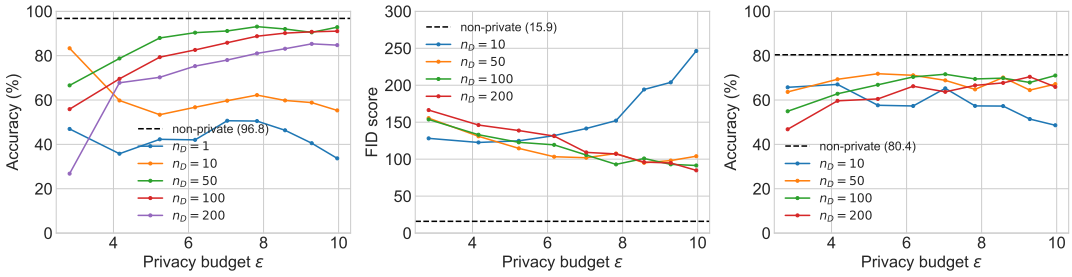
We highlight one convenient property of DP, known as *closure under post-processing*, which says that interacting with a privatized model (e.g., using it to compute gradients on non-sensitive data, drawing samples from it) does not lead to any further privacy violation.

DPSGD. A gradient-based training algorithm can be privatized by employing *differentially private stochastic gradient descent* (DPSGD) [27, 5, 1] 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 [14, 13, 4, 34]. We employ the DPSGD analysis of [23] implemented in Opacus [36].

DPGANs. 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 $\mathcal{G}$ due to post-processing.

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 targeting $(10, 10^{-5})$-DP under different discriminator update frequencies n_D. **(a)** Downstream classification accuracy mirrors FID scores (see Figure 1). Going from to $n_D = 1$ to $n_D = 50$ improves accuracy from $33.7\% \to 92.9\%$. **(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 [7] for $(10, 10^{-5})$-DP generation of FashionMNIST.

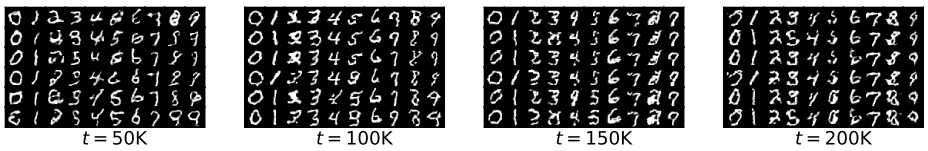


Figure 3: Evolution of samples drawn during training at $n_D = 10$, when targeting $(10, 10^{-5})$-DP. This setting reports its best FID and downstream accuracy at $t = 50$K iterations ($\varepsilon \approx 2.85$). As training progresses, we observe mode collapse alongside deterioration in evaluation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

3.1 Experimental details

Setup. We focus on labelled generation of MNIST and FashionMNIST. To build a strong baseline, we begin from an open source PyTorch [25] implementation[1] of DCGAN [26] that performs well non-privately, and copy their training recipe. We adapt their architecture to our purposes: removing BatchNorm layers (which not compatible with DPSGD) and adding label embedding layers to enable labelled generation. 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.

Privacy implementation. To privatize training, we use Opacus [36] which implements per-example gradient computation and the RDP accounting of [23]. For our baseline setting, we use the following DPSGD hyperparameters: we keep the non-private (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.

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) [18] between $60,000$ generated images and $10,000$ real images from the test set.[2] 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.

3.2 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

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

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

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.

3.3 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?

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 private training for different settings of n_D. Non-privately, discriminator accuracy stays around 60% throughout training. Introducing DP $(n_D = 1)$ leads to a qualitative difference: DP causes discriminator accuracy to drop to 50% immediately, and never recovers. For other 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 on why more steps helps:

- 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.

5 Better generators via better discriminators

Our proposed explanation in Section 4 provides a suggestion for improving GAN training: effectively use our privacy budget to maximize the number of generator steps taken when the discriminator has a high accuracy. 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 [30, 2, 11]. GAN training is typically conducted with small batch sizes (for example, DCGAN uses $B = 128$, which

[3] This observation has been reported in [24], 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.

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.

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 $B = 512$.

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, which co-occurs with degradation of results. In this scenario, we want to take more steps to improve the discriminator. However, using a large discriminator update frequency at the beginning of training is wasteful – 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 on real data 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 fake samples only (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 results of a much more extensive hyperparameter search.

5.3 Comparison with state of the art

Table 1 summarizes our best experimental settings. In the low privacy/high ε regime, most of our results are dramatically better than prior work[4] – 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, improvements 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 for $\varepsilon = 10$, and Figures 7 and 8 for $\varepsilon = 1$.

6 Discussion and related work

DP generative modelling. Differentially private GANs (DPGANs) were introduced by Xie et al. [35]. Subsequent works proposed alternative privatizations schemes for GANs settings [29, 20, 22, 8, 33]; they did not consider a significantly larger number of discriminator steps per generator step. Other private generative models include VAEs [10], maximum mean discrepancy [17], Sinkhorn divergences [7], energy-based models [9], and normalizing flows [32]. We show that a well-tuned DPGAN competes with or outperforms these approaches.

[4] We do not compare with two recent works on private generative models [9, 19], as we believe there are gaps in their privacy analyses. This has been confirmed by the authors of [19], and the sketch of an argument regarding non-privacy of [9] has been shared with us by others [3].

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[5]	[8]	179.16	63	243.80	50
		[22]	304.86	80.11	433.38	60.98
	GS-WGAN	[8]	61.34	80	131.34	65
	PATE-GAN	[22]	253.55	66.67	229.25	62.18
	G-PATE	[22]	150.62	80.92	171.90	69.34
	Datalens	[33]	173.50	80.66	167.68	70.61
	DP-MERF	[7]	116.3	82.1	132.6	**75.5**
	DP-Sinkhorn	[7]	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	[22]	470.20	40.36	472.03	10.53
	GS-WGAN	[22]	489.75	14.32	587.31	16.61
	PATE-GAN	[22]	231.54	41.68	253.19	42.22
	G-PATE	[22]	153.38	58.80	214.78	58.12
	Datalens	[33]	186.06	71.23	194.98	64.78
	DP-MERF	[19]	118.3	80.5	**102.1**	**74.6**
	DP-MERF	[31][6]	-	80.7	-	73.9
	DP-HP	[31][6]	-	**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. For downstream accuracy, we report the best accuracy among classifiers they use, and compare against our CNN classifier accuracy.

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 [7, 17, 31], 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 [12, 21, 37], 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 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 [28] find that private GAN-based approaches fail to preserve even basic statistics in these settings, we believe that our techniques may yield similar improvements.

Conclusion. 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.

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

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

References

[1] 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, pages 308–318, New York, NY, USA, 2016. ACM.

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

[3] Anonymous. Personal communication, 2022. Anonymous for double blind.

[4] 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, pages 6277–6287. Curran Associates, Inc., 2018.

[5] 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, pages 464–473, Washington, DC, USA, 2014. IEEE Computer Society.

[6] 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.

[7] 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.

[8] 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.

[9] 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)*, pages 8387–8396, 2022.

[10] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaafar, and Haojin Zhu. Differentially private data generative models. *CoRR*, abs/1812.02274, 2018.

[11] 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.

[12] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *CoRR*, abs/2204.13650, 2022.

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

[14] 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, pages 486–503, Berlin, Heidelberg, 2006. Springer.

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

[16] 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.

[17] 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)*, pages 1819–1827, 2021.

[18] 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.

[19] Dihong Jiang, Guojun Zhang, Mahdi Karami, Xi Chen, Yunfeng Shao, and Yaoliang Yu. Dp^2-vae: Differentially private pre-trained variational autoencoders. *CoRR*, abs/2208.03409, 2022.

[20] 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.

[21] 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.

[22] 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)*, pages 2965–2977, 2021.

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

[24] Marcel Neunhoeffer, Steven Wu, and Cynthia Dwork. Private post-gan boosting. In *9th International Conference on Learning Representations (ICLR'19)*, 2021.

[25] 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.

[26] 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.

[27] 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, pages 245–248, Washington, DC, USA, 2013. IEEE Computer Society.

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

[29] 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)*, pages 98–104, 2019.

[30] 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.

[31] 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)*, pages 22300–22324, 2022.

[32] Chris Waites and Rachel Cummings. Differentially private normalizing flows for privacy-preserving density estimation. *CoRR*, abs/2103.14068, 2021.

[33] 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*, pages 2146–2168, 2021.

[34] 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, pages 1226–1235. JMLR, Inc., 2019.

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

[36] 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.

[37] 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.

A Extra figures

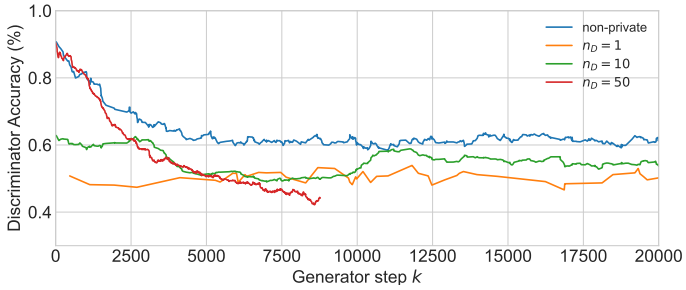


Figure 4: Discriminator accuracy before each generator step. While non-privately the discriminator maintains 60% accuracy, the private discriminator with $n_D = 1$ is effectively a random guess. Increasing the number of discriminator steps recovers the discriminator's advantage early on.

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
  $S_t \sim \text{PoissonSample}(D, q)$ ▷ 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
  $\phi_{t+1} \leftarrow \texttt{OptD}(\phi_t, \widehat{g}_{\theta_t})$ and $t \leftarrow t + 1$
  if n_D divides t **then** ▷ perform $\mathcal{G}$ update every n_D steps
    $\widetilde{S}'_t \sim \mathcal{G}(\cdot; \theta_k)^B$
    $g_{\theta_k} \leftarrow \frac{1}{B} \sum_{(\widetilde{x},\widetilde{y})\in \widetilde{S}'_t} \nabla_{\theta_k}(-\log(\mathcal{D}(\widetilde{x}, \widetilde{y}; \phi_t)))$
    $\theta_{k+1} \leftarrow \texttt{OptG}(\theta_k, g_{\theta_k})$ and $k \leftarrow k + 1$
  end if
end while
$\varepsilon \leftarrow \text{PrivacyAccountant}(T, \sigma, q, \delta)$ ▷ compute privacy budget spent
Output: Final $\mathcal{G}$ parameters θ_k. (ε, δ)-DP guarantee.

B Generated samples

We provide a few non-cherrypicked samples from MNIST and FashionMNIST at $\varepsilon = 1$ and $\varepsilon = 10$.

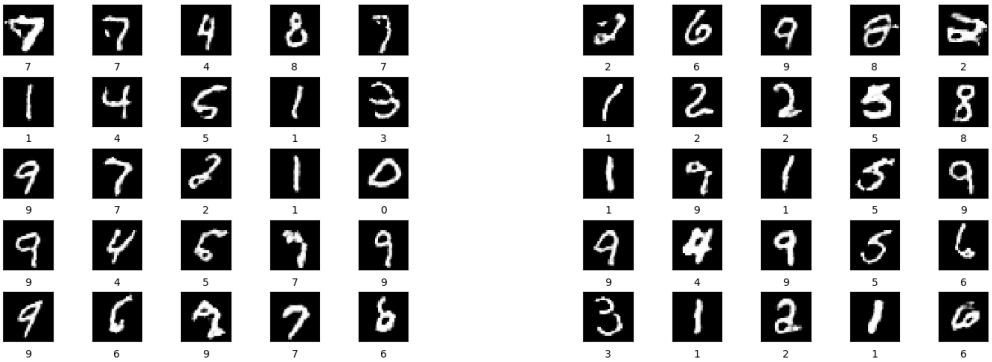


Figure 5: Some non-cherrypicked MNIST samples from our method, $\varepsilon = 10$.

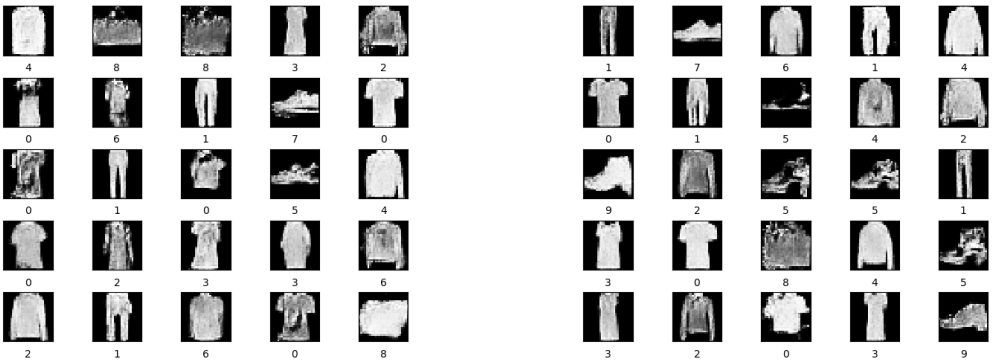


Figure 6: Some non-cherrypicked FashionMNIST samples from our method, $\varepsilon = 10$.

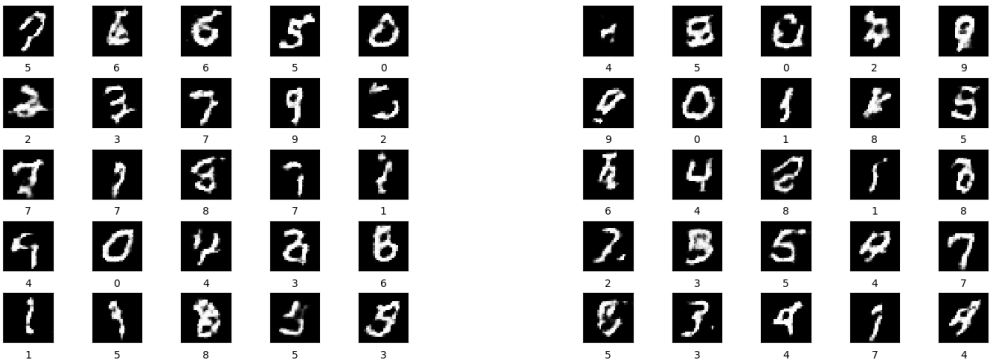


Figure 7: Some non-cherrypicked MNIST samples from our method, $\varepsilon = 1$.

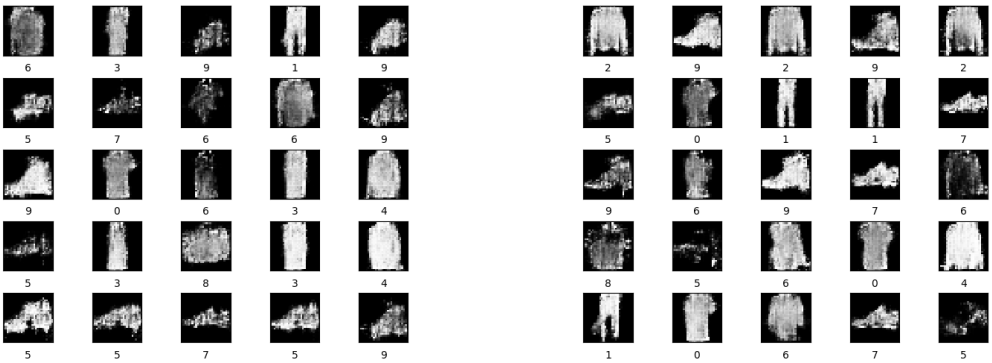


Figure 8: Some non-cherrypicked FashionMNIST samples from our method, $\varepsilon = 1$.