DP-LFlow: Differentially Private Latent Flow for Scalable Sensitive Image Generation

Dihong Jiang [1 2] **Sun Sun** [1 3]

Abstract

Differentially private generative model (DPGM) is designed to generate data that are distributionally similar to the original sensitive data yet with differential privacy (DP) guarantees. While GAN attracts major attention, existing DPGMs based on flow generative models are limited and only developed on low-dimensional tabular datasets. The capability of *exact* density estimation makes the flow model exceptional especially when density estimation is of interest. In this work, we will first show that it is challenging (or even infeasible) to train a DP-flow via DP-SGD, i.e. the workhorse algorithm for private deep learning, on high-dimensional image sets with acceptable utility, and then we give an effective solution by reducing the generation from the pixel space to a lower dimensional latent space. We show the effectiveness and scalability of the proposed method via extensive experiments. Notably, our method is scalable to high-resolution image sets, which is rarely studied in related works.

1. Introduction

Large-scale datasets (Deng et al., 2009; Lewis et al., 2004; Bennett et al., 2007) facilitate the great success of modern machine learning (ML) systems. However, privacy concerns arise when sensitive data (e.g. face images) are involved in the training. Among various privacy-preserving techniques, differential privacy (DP) (Dwork, 2006) is recognized as a rigorous quantization of privacy, which becomes the gold standard in the current ML community.

DPGM aims to synthesize data that are distributionally similar to the private data while satisfying DP guarantees. Therefore, DPGMs can (1) serve as a proxy for releasing private data and can (2) generate data for private data analysis tasks without incurring further privacy cost, as ensured by the post-processing theorem (Dwork et al., 2014).

Generative adversarial network (GAN) (Goodfellow et al., 2014) attracts the most attention in developing DPGMs (Xie et al., 2018; Torkzadehmahani et al., 2019; Jordon et al., 2019; Long et al., 2021; Augenstein et al., 2020; Chen et al., 2020). In contrast, the DPGMs based on the normalizing flow are relatively limited (Waites & Cummings, 2021; Lee et al., 2022). The capability of the *exact* density computation makes flow models particularly useful when density is of interest in some applications, e.g. anomaly detection in a privacy-preserving manner (Waites & Cummings, 2021). However, existing DP-flow works are restricted to tabular datasets (with lower dimensions).

The core steps of DP-SGD (Abadi et al., 2016), i.e. the leading algorithm for training a DP deep learner, are gradient clipping and noise addition. Briefly, let p denote the model dimension, the noise introduced by DP-SGD is $\mathbf{z} \sim \mathcal{N}(\mathbf{0}_p, \sigma^2 C^2 \mathbb{I}_{p\times p})$, where C is the gradient clipping bound and σ is a noise multiplier. Apparently, $\mathbb{E}[\|\mathbf{z}\|_2^2] = p\sigma^2C^2 \propto p$, i.e. the model utility may not be preserved with a large model size under DP-SGD. For example, Yu et al. (2021) show that the gradient will be submerged in the added noise (by DP-SGD) when the model becomes larger on a series of ResNet (He et al., 2016) variants. This utility drop could become more pronounced under strong privacy guarantees (e.g. $\epsilon = 1$).

Training a normalizing flow with DP-SGD on image set seems ostensibly easy, but any ML researcher/engineer will encounter the two non-trivial empirical challenges, as also pointed out by Lee et al. (2022): (1) *batch normalization (BN) challenge*: flow models usually apply a batch normalization (BN) layer in each block to boost the performance, where the per-example gradient (in DP-SGD) is not available; (2) *model complexity (MC) challenge*: flow models generally consist of repetitive blocks of invertible transformations with a large depth, which results in higher model complexity compared to other generative models. We explore the challenges with two SoTA flow models, i.e. RealNVP (Dinh et al., 2017) and Glow (Kingma & Dhariwal, 2018). Figure 1 show that the generations from a

[1]Cheriton School of Computer Science, University of Waterloo [2]Vector Institute [3]National Research Council. Correspondence to: Dihong Jiang <dihong.jiang@uwaterloo.ca>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

Figure 1. Training a RealNVP (∼150 MegaBytes (MB)) on CelebA (a) non-privately with BN layers, (b) non-privately without BN layers, (c) under $(10, 10^{-5})$-DP without BN layers

DP-RealNVP (subfigure (c)) are largely submerged in the noise. In contrast, Glow uses activation normalization as an alternative to BN, thus per-example gradient is computable (i.e. no BN challenge for Glow). However, we trained Glow with DP-SGD but end up with null synthesis when targeting $(10, 10^{-5})$-DP. Both RealNVP and Glow suffer from the MC challenge.

Our contributions can be summarized as follows:

- We explore the challenge of training a DP flow via DP-SGD on image sets, and propose an efficient and effective solution, i.e. differentially private latent flow (DP-LFlow), by reducing the training of flow from the full pixel space to a lower-dimensional latent space, which is more resilient to the noise perturbation (by DP-SGD).
- Training DPGM on high-resolution images (256×256 pixels and beyond) is extraordinarily challenging due to the inevitably increased model dimension, and to our best knowledge, none of the existing related works attempt to do so. In this work, we will show that DP-LFlow is also scalable to the high-resolution image (256×256) generation with DP constraints.
- DP-LFlow yields state-of-the-art (SoTA) performance on model utility on widely compared image benchmarks, with more robustness and scalability on different datasets (gray-scale and RGB) and different DP constraints.

We defer the preliminary to Appendix C and related works to Appendix D.

2. Method: DP-LFlow

As shown in (Yu et al., 2021), a smaller ResNet is more resilient to DP training. The intuition is that the model utility will saturate as the model complexity increases. We first confirm this insight for DPGMs (see Appendix G). In fact, shrinking the model size under the DP training will benefit from the following aspects: (1) smaller models are more resilient to (larger) noises (associated with strong DP guarantees); (2) a significant training time overhead remains a notable challenge for DP-SGD (Subramani et al., 2021) due to the gradient clipping and randomization. Smaller models could facilitate more efficient DP training.

Latent Flow: However, a too-simple model is undesired either. We aim to design a model that is small yet expressive enough so that we can achieve a better privacy-utility trade-off with DP-SGD. Inspired by the recent latent diffusion model (Rombach et al., 2022) that achieves SoTA text-to-image generation performance via reducing the diffusion process from the raw input space to a lower dimensional latent space, we propose to train a normalizing flow in a similar manner by simultaneously minimizing the reconstruction loss of the autoencoder and the negative log-likelihood of the flow. As shown in Rombach et al. (2022), the semantic meaning of most images still remains after aggressive compression, thus allowing us to train a flow in an aggressively trimmed latent space, which avoids unnecessary and expensive computation on full input dimensions. It is worth mentioning that latent flow is also not sensitive to BN layers (e.g. for RealNVP), i.e. the utility of latent RealNVP is slightly reduced by removing the BN layer, which validates the use of DP-SGD for latent flow models.

Partitioning Dataset: Current SOTA methods tend to apply conditional generative models, where the label is encoded in the model as part of the input, thus the noise perturbation also distorts the label information, which is unnecessary. We run DP-SGD on the proposed model under the conditional setting (autoencoder + conditional flow), and observe that the label information is largely distorted when $\epsilon = 1$. To circumvent the perturbation to labels, we propose to *partition the dataset according to labels*, train unconditional generative models on each of the subsets, and release the union of all unconditional generators as the resulting model. The partitioning is also beneficial for shrinking the model size, as each generator is only interactive with a sole data modality instead of multi-modalities. Adapted from proposition 2.5 in (Li et al., 2016), the DP guarantee for the union can be derived from the parallel composition, by extending the original parallel composition theorem (Theorem C.2) from ϵ-DP notion to (ϵ, δ)-DP notion. The proof can be found in Appendix A.

Theorem 2.1. *Let $\mathcal{M}_i$ $(i = 1, 2, \ldots, k)$ be k DP mechanisms, and each $\mathcal{M}_i$ satisfies (ϵ_i, δ_i)-DP. Given a deterministic partitioning function f, let $D_1, D_2, \ldots, D_k$ be the disjoint partitions by executing f on D. Releasing $\mathcal{M}_1(D_1), \ldots, \mathcal{M}_k(D_k)$ satisfies $(\max_{i\in\{1,2,\ldots,k\}} \epsilon_i, \max_{i\in\{1,2,\ldots,k\}} \delta_i)$-DP.*

For simplicity, we set ϵ_i the same as the target ϵ for all sub-models in the experiment.

3. Experiments

In this section, we evaluate and compare DP-LFlow against SoTA baselines through extensive experiments. RealNVP is used as the flow model in DP-LFlow, as it yields better performance in practice. Implementation details and neural network configurations are given in the Appendix E.

3.1. Experimental Setup

Datasets: We consider three widely used image datasets, i.e. MNIST (LeCun et al., 1998), Fashion MNIST (Xiao et al., 2017), and CelebA (Liu et al., 2015), as well as one high-resolution RGB datasets (CelebA-HQ (Karras et al., 2018), for our presentation only). For MNIST and Fashion MNIST, we condition on 10 respective labels. For CelebA and CelebA-HQ, we condition on gender. Details of the datasets are given in the Appendix B.

Evaluation Tasks & Metrics: We evaluate and compare DPGMs by two metrics via 60k generated images:

- Fréchet Inception Distance (FID) (Heusel et al., 2017).
- Classification accuracy. We train three different classifiers, e.g. logistic regression (LR), multi-layer perceptron (MLP), and convolutional neural network (CNN), on generated images, then test the classifier on real images, where the performance is measured by the classification accuracy. We take 5 runs and report the average.

SoTA Baselines: DP-CGAN (Torkzadehmahani et al., 2019), DP-MERF (Harder et al., 2021), Datalens (Wang et al., 2021), PATE-GAN (Jordon et al., 2019), G-PATE (Long et al., 2021), GS-WGAN (Chen et al., 2020), DP-Sinkhorn (Cao et al., 2021).

3.2. Comparison with SoTA Baselines

The proposed DP-LFlow is compared with SoTA baselines through extensive qualitative and quantitative experiments on both grayscale and RGB image datasets.

Existing works perform reasonably when $\epsilon = 10$ (Figure 7). Nevertheless, as shown in Table 1, DP-LFlow achieves significant quantitative improvement.

When $\epsilon = 1$, Figure 2 and Figure 3 show that the existing works are not readily amenable to a small ϵ such as 1. In contrast, DP-LFlow exhibits significant visual improvement on all three datasets, which is verified by superior numerical performance in Table 2.

3.3. Generating High-resolution Images under DP

To our best knowledge, the highest resolution image dataset used in related works is CelebA downsampled at 64×64 pixels (Long et al., 2019). Here we consider a real high-resolution dataset CelebA-HQ in 256×256. Note that now the input dimension increases from $1 \times 28 \times 28 = 784$ (for MNIST) to $3 \times 256 \times 256 \approx 1.97 \times 10^5$, which means that the model generally has to drastically scale up to adequately learn the input distribution, leading to a significant challenge for training a DPGM with DP-SGD.

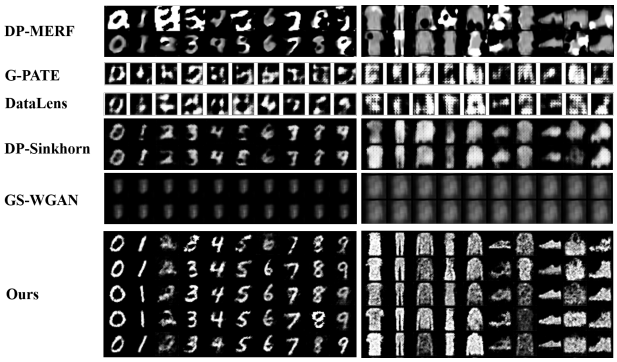

Figure 2. Qualitative comparison on MNIST and Fashion MNIST under $(1, 10^{-5})$-DP.

Figure 3. Qualitative comparison on CelebA conditioned on gender under $(1, 10^{-5})$-DP. Top row: female. Bottom row: male.

Nevertheless, with the help of lower dimensional latent space where we can apply aggressive compression, we are able to restrict the generative model to a size that is suitable for DP training. Figure 4 shows that DP-LFlow is able to produce diverse and recognizable face images with DP constraints on such *high* dimensional input space.

3.4. Out-of-distribution Detection under DP

A natural method to detect out-of-distribution (OOD) input by flow models is to thresholding the density (likelihood) given any input, since the flow model is trained by maximizing the likelihood of in-distribution (InD) data. This idea can be readily extended to latent flow models by checking the density of latent code mapped from the input data. As our sub-models are privately trained on each subset by class, we can immediately conduct the DP intra-dataset OOD detection, by treating each training class as the InD and the rest classes combined as OOD. We

Table 1. Quantitative comparison on MNIST and Fashion MNIST given $(10, 10^{-5})$-DP.

Method	ϵ	MNIST				Fashion MNIST			
		FID ↓	LR Acc ↑	MLP Acc ↑	CNN Acc ↑	FID ↓	LR Acc ↑	MLP Acc ↑	CNN Acc ↑
DP-CGAN	10	179.2	60	60	63	243.8	51	50	46
DP-MERF	10	121.4	79.1	81.1	82.0	110.4	72.3	70.8	73.2
G-PATE	10	150.6	N/A	N/A	80.9	171.9	N/A	N/A	69.3
DataLens	10	173.5	N/A	N/A	80.7	167.7	N/A	N/A	70.6
GS-WGAN	10	61.3	79	79	80	131.3	68	65	65
DP-Sinkhorn ($m = 1$)	10	61.2	79.5	80.2	83.2	145.1	73.0	72.8	70.9
DP-Sinkhorn ($m = 3$)	10	55.6	79.1	79.2	79.1	129.4	70.2	70.2	68.9
Ours	10	**25.4**	**85.1**	**92.4**	**94.8**	**80.8**	**78.1**	**78.4**	**80.7**

Table 2. Quantitative comparison on image datasets given $(1, 10^{-5})$-DP.

Dataset	Metrics	PATE-GAN	DP-MERF	GS-WGAN	G-PATE	DataLens	Ours
MNIST	FID ↓	231.5	118.3	489.8	153.4	186.1	**83.4**
	CNN Acc ↑	41.7	80.5	14.3	58.8	71.2	**88.2**
FMNIST	FID ↓	253.2	**104.2**	587.3	214.8	195.0	143.5
	CNN Acc ↑	42.22	73.1	16.61	58.12	64.8	**76.8**
CelebA	FID ↓	434.5	219.4	437.3	293.2	297.7	**217.7**
	CNN Acc ↑	44.48	57.6	62.9	70.2	70.6	**72.1**

Figure 4. Samples from DP-LFlow trained on CelebA-HQ under $(10, 10^{-5})$-DP. Top 2 rows: female. Bottom 2 rows: male. FID = 328.6, LR classification accuracy = 77.4.

use Area Under Receiver Operating Curve (AUROC) as the evaluation metric. Table 3 indicates that DP-LFlow is able to effectively detect intra-dataset OOD input on both MNIST and Fashion MNIST across all InD classes in a privacy-preserving manner.

Table 3. Differentially private OOD detection results. The evaluation metric is shown by AUROC (higher is better, and 1.0 means InD and OOD likelihood are perfectly separable).

InD class	MNIST		Fashion MNIST	
	$\epsilon = 10$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 1$
0	0.98	0.87	0.90	0.87
1	0.99	0.98	0.97	0.96
2	0.86	0.75	0.90	0.88
3	0.88	0.82	0.91	0.91
4	0.95	0.85	0.87	0.85
5	0.90	0.75	0.91	0.86
6	0.97	0.77	0.84	0.82
7	0.93	0.88	0.97	0.97
8	0.89	0.74	0.86	0.79
9	0.95	0.83	0.94	0.93
Average	0.93	0.82	0.91	0.88

4. Conclusion

Though DP-SGD is currently the workhorse algorithm for training a deep learning model, it remains a big challenge whether it can be reliably applied to large models. In this paper, we first show that training a DP flow via DP-SGD is highly challenging (or even infeasible) with achieving acceptable utility due to a few particular challenges of flow models, and then propose an effective solution, i.e. DP-LFlow, by reducing the flow training from the full input space to a lower dimensional latent space, so that the model is more resilient to (larger) noise perturbation introduced by DP-SGD. Experimental results on widely compared image benchmarks demonstrate the generality and scalability of DP-LFlow on different image spaces (grayscale and RGB) and different DP constraints (weak and strong DP guarantees). Notably, to our best knowledge, DP-LFlow is the first DPGM approach that is amenable to high-resolution image datasets, which further validates its effectiveness and versatility.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308—318, 2016.

Acs, G., Melis, L., Castelluccia, C., and De Cristofaro, E. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1109–1121, 2018.

Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., and y Arcas, B. A. Generative models for effective ml on private, decentralized datasets. In *International Conference on Learning Representations*, 2020.

Bennett, J., Lanning, S., et al. The Netflix prize. In *Proceedings of KDD cup and workshop*, pp. 35, 2007.

Cao, T., Bie, A., Vahdat, A., Fidler, S., and Kreis, K. Don't generate me: Training differentially private generative models with Sinkhorn divergence. In *Advances in Neural Information Processing Systems 21*, 2021.

Chen, D., Orekondy, T., and Fritz, M. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *Advances in Neural Information Processing Systems*, pp. 12673–12684, 2020.

Chen, Q., Xiang, C., Xue, M., Li, B., Borisov, N., Kaarfar, D., and Zhu, H. Differentially private data generative models. arxiv:1812.0227, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Dwork, C. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

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

Harder, F., Adamczewski, K., and Park, M. DP-MERF: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1819–1827, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jordon, J., Yoon, J., and Van Der Schaar, M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

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

Lee, J., Kim, M., Jeong, Y., and Ro, Y. Differentially private normalizing flows for synthetic tabular data generation. In *AAAI Conference on Artificial Intelligence*, 2022.

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5:361–397, 2004.

Li, N., Lyu, M., Su, D., and Yang, W. Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(4):1–138, 2016.

Liew, S. P., Takahashi, T., and Ueno, M. PEARL: Data synthesis via private embeddings and adversarial reconstruction learning. In *International Conference on Learning Representations*, 2022.

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

Long, Y., Lin, S., Yang, Z., Gunter, C. A., and Li, B. Scalable differentially private generative student model via pate. arXiv:1906.09338, 2019.

Long, Y., Wang, B., Yang, Z., Kailkhura, B., Zhang, A., Gunter, C. A., and Li, B. G-PATE: Scalable differentially private data generator via private aggregation of teacher discriminators. In *Advances in Neural Information Processing Systems*, 2021.

McSherry, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 19–30, 2009.

Mironov, I. Rényi differential privacy. In *IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. In *International conference on learning representations*, 2017.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, U. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018.

Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 547–562, 1961.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Subramani, P., Vadivelu, N., and Kamath, G. Enabling fast differentially private SGD via just-in-time compilation and vectorization. In *NeurIPS*, 2021.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Torkzadehmahani, R., Kairouz, P., and Paten, B. DP-CGAN: Differentially private synthetic data and label generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 98–104, 2019.

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

Wang, B., Wu, F., Long, Y., Rimanic, L., Zhang, C., and Li, B. Datalens: Scalable privacy preserving training via gradient compression and aggregation. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 2146–2168, 2021.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747, 2017.

Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. Differentially private generative adversarial network. arxiv:1802.06739, 2018.

Yu, D., Zhang, H., Chen, W., and Liu, T.-Y. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021.

A. Proof

Theorem 2.1. *Let $\mathcal{M}_i$ ($i = 1, 2, \ldots, k$) be k DP mechanisms, and each $\mathcal{M}_i$ satisfies (ϵ_i, δ_i)-DP. Given a deterministic partitioning function f, let $D_1, D_2, \ldots, D_k$ be the disjoint partitions by executing f on D. Releasing $\mathcal{M}_1(D_1), \ldots, \mathcal{M}_k(D_k)$ satisfies $(\max_{i\in\{1,2,\ldots,k\}} \epsilon_i, \max_{i\in\{1,2,\ldots,k\}} \delta_i)$-DP.*

Proof. Without loss of generality, given two neighboring datasets D and D', assume that D contains one more element than D'. Executing f on D and D', we have partitions $D_1, D_2, \ldots, D_k$ and $D'_1, D'_2, \ldots, D'_k$, respectively. There exists j such that (1) D_j contains one more element than D'_j, and (2) $D_s = D'_s$ for $s = 1, 2, \ldots, k$ and $s \neq j$. Denote $\mathcal{M}_1(D_1), \ldots, \mathcal{M}_k(D_k)$ by $\mathcal{M}(D)$. Since the subsets are disjoint from each other, running k algorithms on each subset is independent from each other. For any sequence $t = (t_1, t_2, \ldots, t_k)$ of outputs of $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where $t_i \in Range(\mathcal{M}_i)$, we have:

$$
\begin{aligned}
\Pr[\mathcal{M}(D) = t] &= \Pr[\mathcal{M}_1(D_1) = t_1 \wedge \mathcal{M}_2(D_2) = t_2 \wedge \ldots \wedge \mathcal{M}_k(D_k) = t_k] && (1)\\
&= \Pr[\mathcal{M}_j(D_j) = t_j] \prod_{s=1,2,\ldots,k,s\neq j} \Pr[\mathcal{M}_s(D_s) = t_s] && (2)\\
&\leq \left(\exp(\epsilon_j)\Pr[\mathcal{M}_j(D'_j) = t_j] + \delta_j\right) \prod_{s=1,2,\ldots,k,s\neq j} \Pr[\mathcal{M}_s(D'_s) = t_s] && (3)\\
&= \exp(\epsilon_j) \prod_{i=1,2,\ldots,k} \Pr[\mathcal{M}_i(D'_i) = t_i] + \delta_j \prod_{s=1,2,\ldots,k,s\neq j} \Pr[\mathcal{M}_s(D'_s) = t_s] && (4)\\
&= \exp(\epsilon_j)\Pr[\mathcal{M}(D') = t] + \delta_j \prod_{s=1,2,\ldots,k,s\neq j} \Pr[\mathcal{M}_s(D'_s) = t_s] && (5)\\
&\leq \exp(\epsilon_j)\Pr[\mathcal{M}(D') = t] + \delta_j && (6)\\
&\leq \exp(\max_{i=1,2,\ldots,k} \epsilon_i)\Pr[\mathcal{M}(D') = t] + \max_{i=1,2,\ldots,k} \delta_i && (7)
\end{aligned}
$$

□

B. Datasets

We briefly introduce the public datasets and associated preprocessing. Image size is shown in $\#channels \times height \times width$. Images are normalized to the range of $[0, 1]$.

MNIST (LeCun et al., 1998) & Fashion MNIST (Xiao et al., 2017): MNIST contains hand-written digits images, whereas Fashion MNIST contains cloth and shoe images. Images in both datasets are single-channel, in the size of $1 \times 28 \times 28$, and have 10 classes. We adopt the official training and test split. 10k images from the training split are randomly held out as the validation set.

CelebA (Liu et al., 2015): CelebA is a dataset including face images of celebrities. Each image is in the size of $3 \times 178 \times 218$ and has 40 binary attributes. All images are cropped to $3 \times 178 \times 178$, and then resized to $3 \times 32 \times 32$. We also adopt the official training, validation and test split, but randomly select 50k images of each gender from the training split as our training set.

CelebA-HQ (Karras et al., 2018): CelebA-HQ is a high-quality version of CelebA, which is commonly recognized as a high-resolution image set. It consists of 30k images in total. We download a gender conditioned split (where images are resized to $3 \times 256 \times 256$) by following this link. 1999 images are randomly held out from the training split as the valiation set.

C. Preliminary

In this section, we recall background knowledge in differential privacy.

Algorithm 1 Gradient perturbation in DP-SGD

Input: Private training set $X = \{\mathbf{x}_i\}_{i=1}^N$, loss function $\mathcal{L}(\cdot)$, batch size B, noise multiplier σ, gradient clipping bound C, model parameter θ
for $i = 1$ **to** B **do**
  $g_\theta(\mathbf{x}_i) = \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta)$
  $g_\theta(\mathbf{x}_i) = g_\theta(\mathbf{x}_i) \cdot \min\left(1, \frac{C}{\|g_\theta(\mathbf{x}_i)\|_2}\right)$
end for
$\tilde{g}_\theta = \frac{1}{B}\left[\sum_{i=1}^B g_\theta(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbb{I})\right]$

C.1. Differential Privacy

Differential privacy is widely regarded as a rigorous quantization of privacy, which upper-bounds the deviation in the output distribution of a randomized algorithm given an incremental deviation in the input. Formally, we have the following definition:

Definition C.1 ((ϵ, δ)-DP (Dwork et al., 2014))**.** A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $D, D' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta \tag{8}$$

where adjacent inputs (a.k.a. neighbouring datasets) only differ in one entry. Particularly, when $\delta = 0$, we say that $\mathcal{M}$ is ϵ-DP.

There is a convenient parallel composition theorem for ϵ-DP mechanisms:

Theorem C.2 (Parallel composition theorem of ϵ-DP, (McSherry, 2009))**.** *Let $\mathcal{M}_i$ ($i = 1, 2, \ldots, k$) be k DP mechanisms, and each $\mathcal{M}_i$ satisfies ϵ_i-DP. Given a deterministic partitioning function f, let $D_1, D_2, \ldots, D_k$ be the disjoint partitions by executing f on D. Releasing $\mathcal{M}_1(D_1), \ldots, \mathcal{M}_k(D_k)$ satisfies $\max_{i \in \{1,2,\ldots,k\}} \epsilon_i$-DP.*

We will extend the above parallel composition to the (ϵ, δ)-DP notion in Section 2.

A famous theorem, i.e. post-processing theorem, which is utilized by existing works (as well as ours) for proving DP guarantee of a published model, is given by:

Theorem C.3 (Post-processing theorem, (Dwork et al., 2014))**.** *If $\mathcal{M}$ satisfies (ϵ, δ)-DP, $F \circ \mathcal{M}$ will satisfy (ϵ, δ)-DP for any function F with $\circ$ denoting the composition operator.*

Sampling from a DPGM is independent of training data, thus can be viewed as a post-processing step and does not breach the DP guarantee.

Rényi differential privacy (RDP) extends ordinary DP using Rényi's α divergence (Rényi, 1961) and provides tighter and easier composition property than the ordinary DP notion, thus we adopt RDP to accumulate the privacy cost.

C.2. DP-SGD

Within predetermined training iterations, for each iteration DP-SGD (Abadi et al., 2016) subsamples a batch from the private training set, clip and perturb the gradient as in Algorithm 1, and optimize the model with privatized gradient $\tilde{g}_\theta$. As mentioned earlier, the norm of the Gaussian noise introduced at line 4 in Algorithm 1 will scale linearly with the model dimension, thus will generally degrade the utility of large models.

C.3. Rényi differential privacy (RDP)

Rényi differential privacy (RDP) extends ordinary DP using Rényi's α divergence (Rényi, 1961) and provides tighter and easier composition property than the ordinary DP notion, thus we adopt RDP to accumulate the privacy cost. Formally, we recall:

Definition C.4 ((α, ϵ)-RDP (Mironov, 2017))**.** A randomised mechanism $\mathcal{M}$ is (α, ϵ)-RDP if for all adjacent inputs D, D',

Rényi's α-divergence (of order $\alpha > 1$) between the distribution of $\mathcal{M}(D)$ and $\mathcal{M}(D')$ satisfies:

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) := \frac{1}{\alpha-1}\log \mathbb{E}_{Z\sim Q}\left(\frac{P(Z)}{Q(Z)}\right)^\alpha \leq \epsilon, \tag{9}$$

where P and Q are the density of $\mathcal{M}(D)$ and $\mathcal{M}(D')$, respectively (w.r.t. some dominating measure μ).

Importantly, a mechanism satisfying (α, ϵ)-RDP also satisfies$(\epsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$-DP for any $\delta \in (0, 1)$.

Conveniently, RDP is linearly composable:

Theorem C.5 (Sequential composition of RDP (Mironov, 2017))**.** *If mechanism $\mathcal{M}_i$ satisfies (α, ϵ_i)-RDP for $i = 1, 2, \ldots, k$, then releasing the composed mechanism $(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ satisfies $(\alpha, \sum_{i=1}^k \epsilon_i)$-RDP.*

We also adopt the Gaussian mechanism for achieving RDP:

Definition C.6 (Gaussian mechanism for RDP (Dwork et al., 2014; Mironov, 2017))**.** Let $f : \mathcal{D} \to \mathbb{R}^p$ be an arbitrary p-dimensional function with sensitivity:

$$\Delta_2 f = \max_{D,D'} \|f(D) - f(D')\|_2 \tag{10}$$

for all adjacent datasets $D, D' \in \mathcal{D}$. The Gaussian mechanism $\mathcal{M}_\sigma$ perturb the output of f with Gaussian noise:

$$\mathcal{M}_\sigma = f(D) + \mathcal{N}(0, \sigma^2 \cdot \mathbb{I}) \tag{11}$$

where $\mathbb{I}$ is identity matrix. Then, $\mathcal{M}_\sigma$ satisfies $(\alpha, \frac{\alpha(\Delta_2 f)^2}{2\sigma^2})$-RDP.

DP-SGD tracks the total privacy consumption as follows: (1) for each training iteration, compute the RDP privacy cost for a subsampled batch where Gaussian mechanism is applied; (2) compose RDP mechanisms over training iterations; (3) convert RDP back to (ϵ, δ)-DP. The implementation details are given below.

C.4. Flow-based Generative Models

We briefly recap the flow generative models. Flow models learn a bijective map T between a simple prior distribution q_0 (e.g. Gaussian) and the target distribution q: $\mathbf{z} \sim q_0 \Leftrightarrow \mathsf{T}(\mathbf{z}) \sim q$. Through the change-of-variable formula, the log-likelihood of input is tractable:

$$q_\mathsf{T}(\mathbf{x}) = q_0(\mathbf{z})/|\mathsf{T}'(\mathbf{z})| \tag{12}$$

where $\mathbf{z} = \mathsf{T}^{-1}(\mathbf{x})$. Parameterize the bijective map by neural networks, we can train flow models by minimizing the Kullback–Leibler (KL) divergence between the true and estimated distribution:

$$\min_\mathsf{T} \mathbb{D}_{KL}(q(\mathbf{x})||q_\mathsf{T}(\mathbf{x})) = \min_\mathsf{T} \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{q_\mathsf{T}(\mathbf{x})} \mathrm{d}\mathbf{x} \tag{13}$$

$$= \min_\mathsf{T} \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}[-\log q_\mathsf{T}(\mathbf{x})] - \mathbb{H}[q] \tag{14}$$

where $\mathbb{H}[q]$ is the entropy of true distribution. Therefore, training the flow model amounts to minimizing the negative log-likelihood of input.

D. Related work

We categorize related works by approaches:

DP-SGD: The vast majority of related works are realized by training different generative models with DP-SGD. **GAN:** DP-GAN (Xie et al., 2018) first trains GAN with DP-SGD algorithm, where the discriminator is trained with DP-SGD, then the generator is automatically DP as ensured by post-processing theorem. DP-CGAN (Torkzadehmahani et al., 2019) extends the idea into the conditional setting. **VAE:** DP-VaeGM (Chen et al., 2018) trains k VAEs on k classes of private data with the DP-SGD algorithm, and returns the union as generations. This work only focuses on privacy attacks. DP-kVAE

Table 4. Network configurations for different datasets in the experiments. $\#h\_conv$ denotes the number of hidden sizes in the convolutional layers. $\#h\_lin$ denotes the number of hidden sizes in the linear layers. $\#c$ denotes the length of latent code. $\#b$ means the number of blocks in flow.

Dataset	$\#h\_conv$ in encoder	$\#h\_conv$ of decoder	$\#c$	$\#b$	$\#h\_lin$ of flow
MNIST	[32, 64]	[64, 32]	20	9	200
FMNIST	[32, 64]	[64, 32]	20	9	200
CelebA	[64, 128, 256]	[256, 128, 64]	32	9	200
CelebA-HQ	[16, 32, 64, 128, 256, 512]	[512, 256, 128, 64, 32, 16]	64	12	256

(Acs et al., 2018) first partitions the dataset into k clusters by differentially private kernel k-means method, then trains k VAEs on each data cluster with DP-SGD. However, their generation exhibits clear mode-collapse. **Flow:** DP-NF (Waites & Cummings, 2021) directly trains a flow-based model by DP-SGD. DP-HFlow (Lee et al., 2022) designs a fine-grained gradient clipping strategy to increase the signal-to-noise ratio and accelerate the training. However, both works relating to flow models are limited to (low dimensional) tabular datasets.

PATE Mechanism: Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2017; 2018) is another mechanism for learning a DP model, by perturbing the aggregated information from an ensemble of teacher models with noise. PATE-GAN (Jordon et al., 2019) first applies PATE mechanism to GAN, where the discriminator becomes non-differentiable, thus a student discriminator is trained with all teacher ensembles, which is then used to train the generator. G-PATE (Long et al., 2021) sanitizes the aggregated gradients from teacher discriminators to the generator to make the generator DP. However, gradient vectors need to be discretized in each dimension to employ the PATE mechanism that only takes categorical data as input. DataLens (Wang et al., 2021) further improves G-PATE by introducing a three-step gradient compression and aggregation algorithm called TopAgg.

Kernel-based Methods: DP-MERF (Harder et al., 2021) proposes to perturb the kernel mean embeddings of real data through random Fourier features, and train a generator by minimizing the maximum mean discrepancy (MMD) between the noisy embedding of private input and embedding of generation. PEARL (Liew et al., 2022) extends the idea of DP-MERF by introducing an adversarial objective on sampling frequencies, which indicates improvement in performance.

Others: GS-WGAN (Chen et al., 2020) creates a privacy barrier at the output of the generator of a Wasserstein GAN (WGAN), based on the observation that only the generator will be published, thus only the generator needs to be private. DP-Sinkhorn (Cao et al., 2021) adds a privacy barrier in a similar way as GS-WGAN, where a Sinkhorn loss is used as the training objective.

E. Implementation

E.1. Flow models

Our code for flow generative models are adapted from public repos, i.e. Glow and RealNVP. Hyperparameters of the network are selected by comparing the performance on the validation set, and the selection results are given in Table 4.

E.2. Privacy implementation

We use a public repo, i.e. pyvacy, for implementing DP-SGD algorithm, as well as the total privacy calculation. Pyvacy tracks the privacy loss by RDP accountant, which is a PyTorch implementation based on Tensorflow Privacy.

For all datasets we use, we set subsampling rate as 0.1, training iterations as 300, noise multiplier as 1.25 to target $(10, 10^{-5})$-DP and 4.5 to target $(1, 10^{-5})$-DP, respectively. With better evaluation performance on the validation set, gradient clipping norms are set as 0.1 for MNIST and Fashion MNIST, 0.01 for CelebA, and 10 for CelebA-HQ.

E.3. Fréchet Inception Distance (FID)

FID calculates the distance between the feature vectors extracted by InceptionV3 pool3 layer (Szegedy et al., 2016) on real and synthetic samples. Specifically,

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \tag{15}$$

where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ are activations of InceptionV3 pool3 layer of real images and generated images, respectively, and $\text{Tr}(A)$ refers to the trace of a matrix A. Intuitively, a lower FID means the generation X_g is more realistic (or more similar to X_r). We use a PyTorch implementation for computing FID, which will resize images and repeat channels three times for grayscale images to meet the input size requirement.

E.4. Classification task

We follow Cao et al. (2021) for the classifier implementation. We import scikit-learn package for implementation logistic regression classifier (e.g. from sklearn.linear_model import LogisticRegression) with default parameter settings.

The MLP network consists of following layers: linear($input\_dim, 100$) $\rightarrow$ ReLU $\rightarrow$ linear(100, $output\_dim$) $\rightarrow$ Softmax.

The CNN consists of following layers: Conv2d($input\_channels$, 32, kernel_size=3, stride = 2, padding=1) $\rightarrow$ Dropout(p=0.5) $\rightarrow$ ReLU $\rightarrow$ Conv2d(32, 64, kernel_size=3, stride = 2, padding=1) $\rightarrow$ Dropout(p=0.5) $\rightarrow$ ReLU $\rightarrow$ flatten $\rightarrow$ linear($flatten\_dim, output\_dim$) $\rightarrow$ Softmax.

Both MLP and CNN are optimized by Adam with default parameters. All classifiers are trained on synthetic data, and we report test accuracy on real test data as the evaluation metric.

E.5. Baselines

All results of DP-MERF (Harder et al., 2021) are obtained by running their code with default parameters. It is worth mentioning that DP-MERF does not implement on CelebA. We adapt their code on CelebA by using the generative network they designed for SVHN with $16, 8, 8$ channels for three convolutional layers, respectively.

GS-WGAN only implements on $(10, 10^{-5})$-DP. To target $(1, 10^{-5})$-DP, we tried two vanilla variations by tuning parameters of $(10, 10^{-5})$-DP in their code, i.e. increasing noise scale while keeping the rest parameters unchanged, or decreasing the number of iterations while keeping the rest parameters unchanged. We experimentally found that both variations will not generate meaningful images. The former variation is not even able to generate anything on Fashion MNIST, so we instead present the latter variation for comparison.

All other results (e.g. numbers in the tables, generated images) are cited from papers as we specify.

F. Schematic

The schematic workflow of DP-LFlow is shown in Figure 5.

G. Additional results

G.1. VAE size vs. FID

Consider privately training a VAE on MNIST (LeCun et al., 1998) for example. Figure 6 indicates that a smaller VAE generates better images (with lower FID) than larger counterparts (both qualitatively and quantitatively) under the DP training, even though larger VAEs perform better in the non-DP setting.

G.2. Qualitative comparison under $(10, 10^{-5})$

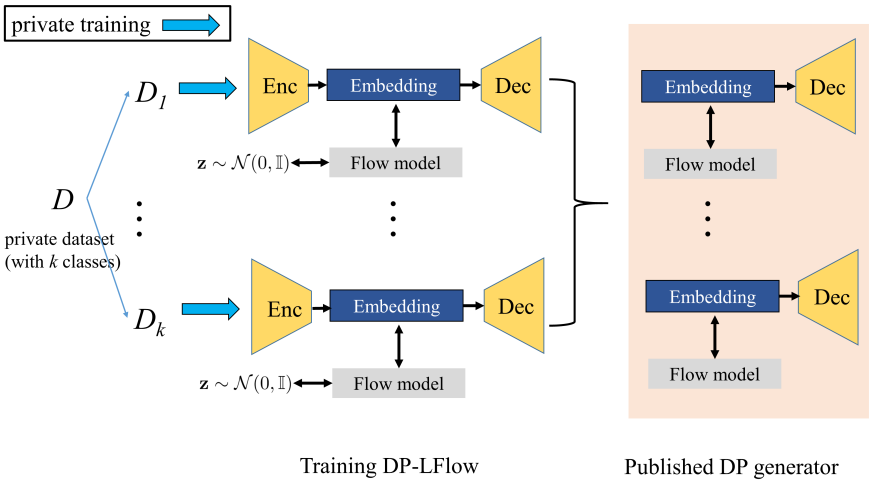

Figure 5. The framework of DP-LFlow.

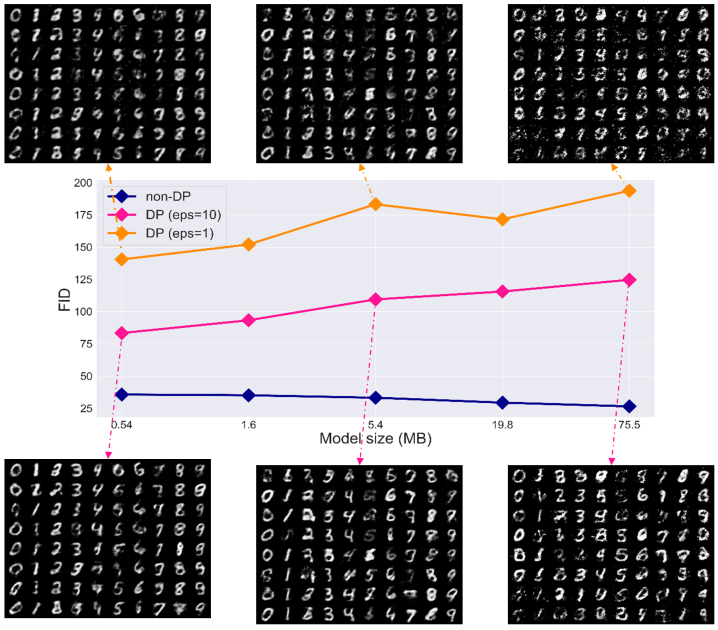

Figure 6. FID vs. VAE size (in MB). We only vary the model complexity, with all the rest training parameters (e.g. subsampling rate, noise multiplier, training iterations) fixed.

Figure 7. Qualitative comparison on MNIST and Fashion MNIST under $(10, 10^{-5})$-DP.