Local Differential Privacy with Entropic Wasserstein Distance

Daria Reshetova [1] **Wei-Ning Chen** [1] **Ayfer Özgür** [1]

Abstract

Local differential privacy (LDP) is a powerful method for privacy-preserving data collection. In this paper, we develop a framework for training Generative Adversarial Networks (GAN) on differentially privatized data. We show that entropic regularization of the Wasserstein distance - a popular regularization method in the literature that has been often leveraged for its computational benefits - can be used to denoise the data distribution when data is privatized by popular additive noise mechanisms, such as Laplace and Gaussian. This combination uniquely enables the mitigation of both the regularization bias and the effects of privatization noise, thereby enhancing the overall efficacy of the model. We analyze the proposed method, provide sample complexity results and experimental evidence to support its efficacy.

1. Introduction

Local differential privacy (Dwork et al., 2006a; Kasiviswanathan et al., 2011) has emerged as a powerful method to provide privacy guarantees on individuals' personal data and has been recently deployed by major technology organizations for privacy-preserving data collection from peripheral devices. In this framework, the user data is locally randomized (e.g. by the addition of noise) before it is transferred to the data curator. Mathematically provable guarantees on the randomization mechanism ensure that any adversary that gets access to the privatized data will be unable to learn too much about the user's personal information. Learning from privatized data, however, requires rethinking machine learning methods to extract accurate and useful population level models from the noisy individual data.

In this paper, we consider the problem of training generative models from locally privatized user data. In recent years, deep learning based generative models, known as Generative Adversarial Networks (GANs) have become a popular framework for learning data distributions and sampling as they have achieved impressive results in various domains (De et al., 2016; Isola et al., 2017; Reed et al., 2016; Ledig et al., 2017). As opposed to traditional methods of fitting a parametric distribution, GANs aim to learn a mapping (usually modeled as a neural network) from a simple known distribution to the unknown data distribution or its empirical approximation. The mapping is set to a minimizer of a chosen distance measure between the generated and target distributions. A popular metric used in practice is the p-Wasserstein distance (see Section 2 for a formal definition), in which case the GAN optimization problem can be written in the following form,

$$\min_{G\in\mathcal{G}} W_p\left(P_{G(Z)}, P_X\right). \tag{1}$$

Here $G(\cdot)$ is called the generator, and comes from a set of functions $\mathcal{G} \subseteq \{G : \mathcal{Z} \to \mathcal{X}\}$ and maps a latent random variable $Z \in \mathcal{Z}$ with some known distribution to a random variable $G(Z) \in \mathcal{X}$, with distribution $P_{G(Z)}$ that is close to some target probability measure P_X in p-Wasserstein distance. The target probability measure P_X is the population distribution from which samples $\{X_i\}_{i=1}^n \sim P_X^{\otimes n}$ are drawn and the optimization problem is solved by replacing P_X in (1) with the empirical distribution Q_X^n of the samples. For example, X_i can represent images taken by users and G represents a generative model for such images.

How can we use the GAN framework above to learn a generative model for P_X when we have only access to samples privatized by an LDP mechanism? Assume now that each X_i represents a sample locally generated at a different user i, and is privatized by a randomized LDP mechanism $M : \mathcal{X} \to \mathcal{Y}$. The learner only observes the privatized samples $\{Y_i = M(X_i)\}_{i=1}^n$. Can we learn a generative model for the true distribution P_X from the privatized samples $Y_i{}_{i=1}^n$? Simply replacing the target distribution P_X in (1) with the empirical distribution Q_Y^n of the observed samples,

$$\min_{G\in\mathcal{G}} W_p^p\left(P_{G(Z)}, Q_Y^n\right), \tag{2}$$

will result in a generative model for $P_Y = M\#P_X$, the push-forward distribution of P_X through the privatization mechanism M, rather than the original distribution P_X.

[1]Department of Electrical Engineering, Stanford University, Location, Country. Correspondence to: Daria Reshetova <resh@stanford.edu>.

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

In other words, we will learn to generate samples from the privatized distribution, e.g. noisy images, instead of learning to generate samples from the original (raw) distribution.

In this paper, we show that a simple but non-intuitive modification of the objective in (2) – the addition of an entropic regularization term – allows one to provably learn the original distribution of the samples under de-facto privatization mechanisms such as the local Laplace or Gaussian mechanism. We first show that in the population case when Q_Y^n is replaced by $P_Y = M\#P_X$, the optimal solution G^* of the *entropic* p-Wasserstein GAN is such that $P_{G^*(Z)} = P_X$ (assuming $\mathcal{G}$ is rich enough to generate P_X). Here, p is chosen to match the privatization mechanism used, e.g. $p = 1$ for the Laplace mechanism and $p = 2$ for the Gaussian mechanism. This result shows that the entropic regularization acts as a denoiser for the Gaussian mechanism under the W_2 distance, and the Laplace mechanism under the W_1 distance. We also provide sample complexity results which suggest that the solution of the empirical problem (when P_Y is replaced by Q_Y^n) converges to the population solution at the parametric convergence rate $O(1/\sqrt{n})$.

Our main contributions include:

- *LDP Framework for Wasserstein GANs*: We propose a novel modification to the widely adopted Wasserstein GAN framework that enables it to learn effectively from LDP samples with only one communication round between the data holders and the server. This adaptation, which is both simple and non-intuitive, provides a solution for privacy-preserving learning that does not require any training method modifications.
- *Sample Complexity Bounds*: An essential element of our work involves providing sample complexity bounds. These bounds offer theoretical insights into the performance and scalability of our proposed method, providing a clear understanding of the trade-off between privacy, accuracy, and the volume of data.
- *Empirical Validation:* We supplement our theoretical contributions with a comprehensive set of experiments designed to validate our claims. These experiments demonstrate the efficacy of our approach in practical scenarios and provide empirical evidence of the superior performance of our method.

1.1. Related Work

Estimation, inference and learning problems under local differential privacy constraints have been of significant interest in the recent literature with emphasis on two canonical tasks: discrete distribution and mean estimation (Bassily et al., 2017; Bun et al., 2019; Chen et al., 2021; 2020a; Suresh et al., 2017; Bhowmick et al., 2018; Han et al., 2018). However, insights from these solutions do not extend to learning high-dimensional distributions under LDP constraints.

The exploration of differentially private learning in generative models has primarily been focused on introducing privacy during the training phase, e.g. by adding noise to the gradients during training.(Chen et al., 2020b; Cao et al., 2021; Xie et al., 2018; Zhang et al., 2018; Mansbridge et al., 2020). In contrast, in our framework privatization is achieved at the data level, and the training of the GAN is effectively indistinguishable from the non-private case.

Moreover, our local differential privacy framework is non-interactive, which means that the data is privatized and released only once, no further interaction is expected from the data holders. In contrast, previous approaches (Chen et al., 2020b; Cao et al., 2021; Xie et al., 2018; Zhang et al., 2018; Mansbridge et al., 2020) that introduce privacy in the optimization phase, e.g. DP-SGD, require the model updates to be transmitted back and forth between the data holder and the server at each iteration of the optimization algorithm (Behera et al., 2022) or the raw (unprivatized) data to be transmitted to a server, to which the training algorithm has access. We note that the privacy guarantees achieved under these two different settings, non-interactive/local DP vs. interactive/central DP are not directly comparable. For example, (Kasiviswanathan et al., 2011) showed that achieving privacy in the interactive setting is significantly easier than in the non-interactive setting. In particular, they give an example of a problem that is privately learnable with a polynomial number of samples with interaction but requires an exponential (in terms of dimension) number of samples in the non-interactive setting.

2. Background and Problem Formulation

2.1. Local Differential Privacy

A local randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Z}$ acting on the data domain $\mathcal{X}$, satisfies (ϵ, δ)-(approximate) local differential privacy (DP) (Kasiviswanathan et al., 2011) for $\epsilon \geq 0, \delta \in (0, 1)$ if for any $S \subseteq \mathcal{Z}$ and for any pair of inputs $x, x' \in \mathcal{X}$ it holds that

$$P(\mathcal{A}(x) \in S) \leq e^{\epsilon} P(\mathcal{A}(x') \in S) + \delta \tag{3}$$

LDP ensures that the input to $\mathcal{A}$ cannot be determined from its output with high confidence (determined by ϵ). When $\delta = 0$, we refer to *pure* local differential privacy. One of the most common ways of achieving pure local differential privacy is via the Laplace mechanism.

Laplace Mechanism (Dwork et al., 2006a). For any $\epsilon > 0$ and any function $f : \mathcal{X} \to \mathbb{R}^k$ such that $\|f(x) - f(x')\|_1 \leq \Delta$ for any $x, x' \in \mathcal{X}$, the randomized mechanism $\mathcal{A}(x) = f(x)+(s_1, \ldots, s_k)$ with $s_i \sim \text{Laplace}(0, \Delta/\epsilon)$ independent of $s_j, j \neq i$ is ϵ-DP and is called the Laplace Mechanism. We will call ϵ/Δ the noise scale of the mechanism.

For (ε, δ)-DP with $\delta > 0$, one of the most versatile mechanisms is the Gaussian Mechanism.

Gaussian Mechanism (Dwork et al., 2006b; 2014; Zhao et al., 2019). For any $\epsilon > 0$, $\delta \in (0, 0.5)$, and any function $f : \mathcal{X} \to \mathbb{R}^k$ such that $\|f(x)-f(x')\|_2 \leq \Delta$ for any $x, x' \in \mathcal{X}$, the randomized mechanism $\mathcal{A}(x) = f(x)+(s_1, \ldots, s_k)$ with $s_i \sim \mathcal{N}(0, \sigma^2)$ independent of $s_j, j \neq i$ is called the Gaussian Mechanism and is (ϵ, δ)-DP if

$$\sigma > \frac{c + \sqrt{c^2 + \epsilon}}{\epsilon\sqrt{2}}\Delta, \text{where } c^2 = \ln \frac{2}{\sqrt{16\delta + 1} - 1}. \quad (4)$$

Similar to the Laplacian mechanism, will call σ the noise scale of the Gaussian mechanism.

2.2. Wassserstein GANs

p-Wasserstein distance. Let $p \geq 1$ and $\mathcal{P}(\mathcal{U})$ be the set of all probability measures with support $\mathcal{U} \subseteq \mathbb{R}^d$. Then for $\mathcal{U}, \mathcal{V} \subseteq \mathbb{R}^d$ and $P_U \in \mathcal{P}(\mathcal{U}), P_V \in \mathcal{P}(\mathcal{V})$ – two probability measures on $\mathcal{U}, \mathcal{V}$ with finite p-order moments the p-Wasserstein distance between P_U, P_V (raised to power p) is

$$W_p^p(P_U, P_V) = \inf_{\pi \in \Pi(P_U, P_V)} \mathbb{E}_{(U,V)\sim\pi} \left[\|U - V\|_p^p\right], \quad (5)$$

where $\Pi(P_U, P_V) = \{\pi \in \mathcal{P}(\mathcal{U} \times \mathcal{V}) : \int_{\mathcal{V}} \pi(u, v)dv = P_U(U), \int_{\mathcal{U}} \pi(u, v)du = P_V(v)\}$ is the set of all couplings of P_U and P_V – all joint probability measures with marginal distributions P_U and P_V.

Wasserstein GAN. The main objective of GANs is to find a mapping $G(\cdot)$, called generator, that comes from a set of functions $\mathcal{G} \subseteq \{G : \mathcal{Z} \to \mathcal{X}\}$ (usually modeled as a neural network) and maps a latent random variable $Z \in \mathcal{Z}$ with some known distribution to a variable $X \in \mathcal{X}$ with some target probability measure P_X. Using the p-Wasserstein distance to measure the dissimilarity between the generated $P_{G(Z)}$ and target distribution X leads to the following learning problem of GAN:

$$\min_{G\in\mathcal{G}} W_p^p\left(P_{G(Z)}, P_X\right). \quad (6)$$

Entropic Wasserstein GAN. Entropic regularization to the Wasserstein distance objective has been proposed to make the problem strongly convex and thus solvable in linear time (Peyré et al., 2019). Formally, the entropy-regularized p-Wasserstein distance is defined as

$$W_{p,\lambda}(P_U, P_V) = \inf_{\pi \in \Pi(P_U, P_V)} \mathbb{E}_{(U,V)\sim\pi} \left[\|U - V\|_p^p\right] + \lambda I_\pi(U, V), \quad (7)$$

where $I_\pi(U, V) = \int \log\left(\frac{d\pi(u,v)}{dP_U(u)dP_Y(V)}\right) d\pi(u, v)$ is the mutual information between X, Y under the coupling π. The corresponding GAN objective is the the entropy-regularized p-Wasserstein distance between the generated distribution $G\#P_Z = P_{G(Z)}$ for some latent noise Z and the empirical approximation of target distribution Q_X^n:

$$\min_{G\in\mathcal{G}} W_{p,\lambda}(P_{G(Z)}, Q_X^n), \quad (8)$$

2.3. Wassserstein GANs with LDP Data

Let $M : \mathcal{X} \to \mathcal{Y}$ be a randomized noise-additive privacy preserving mechanism: $Y = M(X) = X + N$, where the noise is sampled from pdf f_N independent of the input X. f_N can be the Laplace pdf for the Laplace mechanism or the Gaussian pdf for the Gaussian mechanism. Let $P_Y = M\#P_X$ denote the distribution of Y, i.e. the push-forward distribution of P_X through the privatization mechanism M. The goal of learning a Wasserstein GAN from privatized samples is to reconstruct $G(Z) \approx X$ in distribution from a sample $S = \{Y\}_{i=1}^n \sim P_Y^{\bigotimes n}$ with empirical distribution $Q_Y^n = \frac{1}{n}\sum_{i=1}^n \delta_{Y_i}$.

3. Main Results

First, we show that solving (8)indeed recovers the target distribution P_X in the population setting, i.e. when one has access to the generating distribution P_Y of the privatized samples, provided that the model class is rich enough to generate the target distribution P_X.

Theorem 1. *Let $X \sim P_X$ and $Y = M(X) = X + N$, where $N = (N_1, \ldots, N_d) \sim f_N$ independent of X and $f_N(x) \propto e^{-\|x\|_p^p/(p\sigma^p)}$ and*

$$G^* = \arg\min_{G\in\mathcal{G}} W_{p,p\sigma^p}(P_{G(Z)}, P_Y). \quad (9)$$

We have:

(i) If $P_X \in \{P_{G(Z)} \mid G \in \mathcal{G}\}$, and then $P_{G^*(Z)} = P_X$.

(ii) If $P_X \notin \{P_{G(Z)} \mid G \in \mathcal{G}\}$, then for $p \in \{1, 2\}$:

$$D_{KL}(P_{G^*(Z)+N}\|P_{X+N}) \leq \min_{G\in\mathcal{G}} W_p^p(P_{G^*(Z)}, P_X), \quad (10)$$

where $D_{KL}(P\|Q) = \int \log \frac{dP}{dQ} dP$ is the KL-divergence.

The theorem indicates that the optimal solution to the GAN optimization problem (9) generates the target distribution P_X. Thus provided that there are enough samples, the generator will output the target distribution. Moreover, when the true data distribution P_X cannot be exactly generated by any model in $\mathcal{G}$, i.e. the approximation error of the class $\mathcal{G}$ is non-zero, the theorem bounds the KL Divergence between the pushforwards of the generated and target distribution. The KL divergence in (9) is sometimes called the smoothed KL divergence between P_X and $P_{G(Z)}$ (Goldfeld et al.,

2020). (10) ensures that if ϵ is the approximation error in p-Wasserstein distance of the class $\mathcal{G}$, then $P_{G*(Z)}$ is ϵ-close to the target distribution P_X in smoothed KL-divergence. The theorem thus justifies using entropic Wasserstein distance as a loss function for LDP additive noise mechanisms. We give the exact settings of Laplace and Gaussian mechanisms in the following corollaries.

Corollary 1. *Under the conditions of Theorem 1, if* $\sup_{x\in\mathcal{X}}\|x\|_1 \le \Delta_1$, $p=1$, *and* $Y=M(X)$ *is the Laplacian mechanism with noise scale* ϵ/Δ_1, *then training a GAN with loss* $W_{1,\epsilon/\Delta_1}(P_{G(Z)},P_Y)$ *is* ϵ*-LDP, and recovers the target distribution:* $P_{G^*(Z)}=P_X$.

Corollary 2. *Under the conditions of Theorem 1, if* $\sup_{x\in\mathcal{X}}\|x\|_2 \le \Delta_2$, $p=2$, *and* $Y=M(X)$ *is the Gaussian mechanism with noise scale* σ *defined in* (4), *then training a GAN with loss* $W_{2,2\sigma^2}(P_{G(Z)},P_Y)$ *is* (ϵ,δ)*-LDP, and recovers the target distribution:* $P_{G^*(Z)}=P_X$.

We next develop sample complexity results for $p=2$ by building on (Reshetova et al., 2021). To formally state the sample complexity results, let us first recall some definitions. A distribution P_X supported on a d-dimensional set $\mathcal{X}$ is σ^2 sub-gaussian for $\sigma\ge 0$ if $\mathbb{E}\exp\left(\|X\|^2/(2d\sigma^2)\right)\le 2$. Let $\sigma^2(X)=\min\{\sigma\ge 0 \big| \mathbb{E}\exp(\|X\|^2/(2d\sigma^2))\le 2\}$, denote the sub-gaussian parameter of the distribution of X. A set of generators $\mathcal{G}$ is said to be star-shaped with a center at 0 if a line segment between 0 and $G\in\mathcal{G}$ also lies in $\mathcal{G}$, i.e.

$$G\in\mathcal{G} \;\Rightarrow \alpha G\in\mathcal{G}, \forall\alpha\in[0,1]. \tag{11}$$

Note that these conditions are not very restricting. For example, the set of all linear generators, the set of linear functions with a bounded norm or a fixed dimension, the set of all L-Lipschitz functions, or neural networks with a relu ($f(x)=\max(0,x)$) activation function at the last layer all satisfy it.

Theorem 2. *(Excess risk) Let* P_Z *and* P_X *be sub-gaussian, the support of* P_X *be* d*-dimensional and the generator set* $\mathcal{G}$ *consist of L-Lipschitz functions, i.e.* $\|G(Z_1)-G(Z_2)\|\le L\|Z_1-Z_2\|$ *for any* $Z_1,Z_2\in\mathcal{Z}$. *and let* $\mathcal{G}$ *satisfy* (11). *If* $Y=M(X)=X+N$ *is the Gaussian mechanism with noise scale* σ *then for*

$$G^*=\arg\min_{G\in\mathcal{G}} W_{2,2\sigma^2}(P_{G(Z)},P_Y), \text{ and}$$

$$G_n=\arg\min_{G\in\mathcal{G}} W_{2,2\sigma^2}(P_{G(Z)},Q_Y^n),$$

where Q_Y^n *is the empirical distribution of* n *i.i.d. samples from* P_Y *it holds that*

$$\mathbb{E}\big[W_{2,2\sigma^2}(P_{G_n(Z)},P_Y)-W_{2,2\sigma^2}(P_{G^*(Z)},P_Y)\big] \le C_d\sigma^2 n^{-1/2}\big(1+(\tau^2(1+\sigma(X)/\sigma)^2)^{\lceil 5d/4\rceil+3}\big),$$

where $\tau=\max\{L\sigma(Z)/\sigma(X),1\}$ *and* C_d *is a dimension dependent constant.*

The generalization error and the distance between the generated and target distributions is thus parametric (of order $1/\sqrt{n}$), which breaks the curse of dimensionality (convergence of order $n^{-\Omega(1/d)}$), often attributed to GANs. However, the rate is still exponential in the dimension. We also observe that the generalization error is approximately linear in σ^2, the privatization noise scale, beyond a certain threshold ($\sigma^2>\sigma(X)^2$)). This implies that convergence for larger σ^2 can be achieved by increasing the number of samples n.

We note that since privatization happens at the data level, the number of optimization rounds is not bounded by the privacy budget, and thus empirical loss minimization can be performed up to the desired accuracy by any optimization method. As opposed to our method, (Chen et al., 2020b; Cao et al., 2021; Xie et al., 2018; Zhang et al., 2018; Mansbridge et al., 2020) constrain the optimizer to DP-SGD and its variants thereby only guaranteeing convergence to a saddle point of the empirical problem (as discussed in (Pichapati et al., 2019)) and introducing bias to the empirical optimization problem. In our method, however, the utility is fully defined by the statistical convergence and is controlled by theorem 2 for the optimal generator G_n (estimated based on the privatized data).

The above result shows that the value of the loss function under the empirical solution G_n converges to the value of the loss function under the population solution G^*. However, this result does not directly relate $P_{G_n(Z)}$ to P_X. Next, we use Theorem 2 to upper bound smoothed KL-divergence between P_X and $P_{G_n(Z)}$.

Corollary 3. *Under the conditions of Theorem 2, if additionally the target distribution can be generated, i.e.* $P_X\in\{P_{G(Z)}\mid G\in\mathcal{G}\}$, *one has*

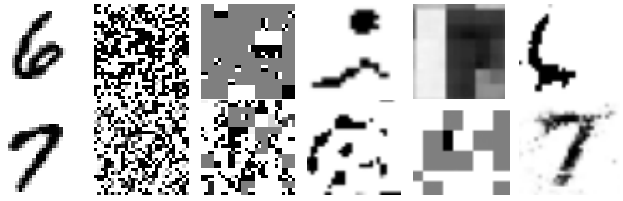
$$\mathbb{E}\big[D_{KL}(P_Y\|P_{G^n(Z)}*\mathcal{N}(0,\sigma^2 I)\big] \le C_d n^{-1/2}\big(1+(\tau^2(1+\sigma(X)/\sigma)^2)^{\lceil 5d/4\rceil+3}\big) \tag{12}$$

Note that the parametric convergence of the smoothed KL-divergence results in the convergence of the Gaussian-smoothed Wasserstein distance (Goldfeld et al., 2020), which is, in turn, a distance metrizing weak convergence similar to W_p.

4. Experimental Results

We conduct our experiments for both Laplace and Gaussian data privatization mechanisms and use the Sinkhorn-Knopp algorithm (Flamary et al., 2021) to approximate the optimal transport plan π in (8). We train our models on MNIST data (LeCun, 1998), consisting of 60000 grayscale images of handwritten digits, we do not use the labels to mimic a fully unsupervised training scenario. The generator model is DCGAN from (Radford et al., 2015). Additional details

and more experiments are provided in the appendix. Here we show that our approach results in much better images, compared to the ones denoised with wavelet transform (Mallat, 1999). The wavelet transform parameters for denoising (the wavelet basis, the level and reconstruction thresholds) were chosen to minimize the average distance between the reconstructed and original image under the particular noise instance, thus providing better results than one would expect in a fully privatized setting. A Wasserstein-GAN (Gulrajani et al., 2017) was trained on the privatized samples as well as the wavelet denoised image. We compare the results to our method (8), denoted entropic WGAN in Figure 1.



original | privatized | wavelet denoising | p-WGAN (1) | denoise+p-WGAN | Entropic WGAN(8)

Figure 1: Image samples privatized with Gaussian $(\epsilon, \delta) = (35, 10^{-4})$ (top) and Laplace mechanism $\epsilon = 196$ (bottom) and generated images from GANs trained on the data.

The results demonstrate that naive denoising with wavelet transform, which is a standard for image denoising, is unable to reconstruct the mnist images privatized with either Gaussian or Laplace noise at the chosen privatization level, and WGAN is not able to learn from either privatized or denoised images. In contrast, the entropic p-WGAN generator learned with the privatized samples was able to learn the distribution far beyond the values of ϵ needed for the wavelet transform reconstruction, demonstrating the efficacy of our method.

Acknowledgements

This work was supported in part by NSF Award #CCF-2213223.

References

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006a.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3): 793–826, 2011.

Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya, and Manuel Gomez Rodriguez. Learning and forecasting opinion dynamics in social networks. In *Advances in Neural Information Processing Systems*, volume 29, pages 397–405. Curran Associates, Inc., 2016.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. Practical locally private heavy hitters. *Advances in Neural Information Processing Systems*, 30, 2017.

Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *ACM Transactions on Algorithms (TALG)*, 15(4):1–40, 2019.

Wei-Ning Chen, Peter Kairouz, and Ayfer Ozgur. Breaking the dimension dependence in sparse distribution estimation under communication constraints. In *Conference on Learning Theory*, pages 1028–1059. PMLR, 2021.

Wei-Ning Chen, Peter Kairouz, and Ayfer Ozgur. Breaking the communication-privacy-accuracy trilemma. *Advances in Neural Information Processing Systems*, 33: 3312–3324, 2020a.

Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International conference on machine learning*, pages 3329–3337. PMLR, 2017.

Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

Yanjun Han, Pritam Mukherjee, Ayfer Ozgur, and Tsachy Weissman. Distributed statistical estimation of high-dimensional and nonparametric distributions. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 506–510. IEEE, 2018.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020b.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34: 12480–12492, 2021.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*, 2018.

Alex Mansbridge, Gregory Barbour, Davide Piras, Christopher Frye, Ilya Feige, and David Barber. Learning to noise: Application-agnostic data sharing with local differential privacy. *arXiv preprint arXiv:2010.12464*, 2020.

Monik Raj Behera, Sudhir Upadhyay, Suresh Shetty, Sudha Priyadarshini, Palka Patel, and Ker Farn Lee. Fedsyn: Synthetic data generation using federated learning. *arXiv preprint arXiv:2203.05931*, 2022.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006b.

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

Jun Zhao, Teng Wang, Tao Bai, Kwok-Yan Lam, Zhiying Xu, Shuyu Shi, Xuebin Ren, Xinyu Yang, Yang Liu, and Han Yu. Reviewing and improving the gaussian mechanism for differential privacy. *arXiv preprint arXiv:1911.12060*, 2019.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Ziv Goldfeld, Kristjan Greenewald, Jonathan Niles-Weed, and Yury Polyanskiy. Convergence of smoothed empirical measures with applications to entropy estimation. *IEEE Transactions on Information Theory*, 66(7):4368–4391, 2020.

Daria Reshetova, Yikun Bai, Xiugang Wu, and Ayfer Özgür. Understanding entropic regularization in gans. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 825–830. IEEE, 2021.

Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779, 2017.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019.

5. Appendix

5.1. Proof of Part (i) of Theorem 1

Proof. We first prove that $P_{G^*(Z)} = P_X$ if $P_X \in \{P_{G(Z)} \mid G \in \mathcal{G}\}$.

Fix some $G \in \mathcal{G}$ and recall that the differential entropy of a random variable U with density μ is $h(U) = -\int \mu(x) \log \mu(x) dx$. Rewriting the mutual information in terms of the differential entropies we get

$$\begin{aligned} W_{p,p\sigma^p}(P_{G(Z)}, P_Y) &= \min_{\pi \in \Pi(P_{G(Z)}, P_Y)} \mathbb{E}_{(G(Z),Y)\sim\pi}[\|G(Z) - Y\|_p^p] + p\sigma^p I_\pi(G(Z), Y) \\ &= \min_{\pi \in \Pi(P_{G(Z)}, P_Y)} \mathbb{E}_{(G(Z),Y)\sim\pi}[\|G(Z) - Y\|_p^p] + p\sigma^p (h(Y) - h(Y \mid G(Z)) \\ &= \min_{\pi \in \Pi(P_{G(Z)}, P_Y)} \mathbb{E}_{(G(Z),Y)\sim\pi}[\|G(Z) - Y\|_p^p] + p\sigma^p (h(Y) - h(Y - G(Z) \mid G(Z)) \end{aligned}$$

Note that the last term on the RHS is upper bounded by $h(Y - G(Z))$ since conditioning cannot increase differential entropy and the equality holds iff $Y - G(Z)$ and $G(Z)$ are independent. Denoting now $D = Y - G(Z)$ results in

$$\begin{aligned} W_{p,p\sigma^p}(P_{G(Z)}, P_Y) &\geq \min_{\pi \in \Pi(P_{G(Z)}, P_Y)} \mathbb{E}_{(G(Z),Y)\sim\pi}[\|D\|_p^p] - p\sigma^p h(D) + p\sigma^p h(Y) \\ &\geq \min_{\pi \in \Pi(P_{G(Z)}, P_Y)} \sum_{i=1}^{d} \mathbb{E}_{(G(Z),Y)\sim\pi}[|D_i|^p] - p\sigma^p h(D_i) + p\sigma^p h(Y), \end{aligned} \tag{13}$$

where we use that $h(D) = h(D_1, \ldots, D_d) \leq \sum_{i=1}^d h(D_i)$, i.e. that the entropy of a vector is maximized iff its components are independent. $h(D_i)$ in the RHS can now be bounded by the maximum entropy of a random variable with a fixed p-th moment. It can be checked that the maximum entropy distribution for $\mathbb{E}|D_i|^p = m_i^p$ is

$$f_{\max, m_i}(x) = \frac{1}{m_i C_N} e^{-|x|^p / p m_i^p},$$

where C_N is the normalization constant that only depends on p. Plugging this into (13) gives

$$\begin{aligned} W_{p,p\sigma^p}(P_{G(Z)}, P_Y) &\geq \min_{m_i > 0} \sum_{i=1}^{d} m_i^p - \sigma^p \log(e m_i^p C_N^p) + p\sigma^p h(Y) \\ &\geq d\sigma^p (p \log(\sigma C_N)) + p\sigma^p h(Y) \\ &= \sigma^p (d - ph(N) + ph(Y)) \end{aligned} \tag{14}$$

where (14) follows from minimizing the RHS over $m_i^p > 0$, which leads to $m_i = \sigma$ and the value of differential entropy of N :

$$\begin{aligned} h(N) = d h(N_i) &= d(\mathbb{E}[|N_i|^p]/(p m_i^p) + \log(m_i C_N)) \\ &= d(1/p + \log(m_i C_N)) = d(1/p + \log(\sigma C_N)) \end{aligned}$$

It is easy to check that the RHS value of (14) is achieved whenever the coupling π is such that $\pi(y \mid G(z)) = f_N(y - G(z))$, which is a feasible coupling if and only if $Y = G(Z) + N$ a.s. Thus, minimizing over $G \in \mathcal{G}$ on both sides gives $P_{G^*(Z)} = P_X$.

□

5.2. Proof of Part (ii) of Theorem 1

We first prove the following lemma, which is used in the proof of Theorem 1 and Corollary 3.

Lemma 5.1. *Let $X \sim P_X, G \sim P_G$ and $Y = M(X) = X + N$, where $N = (N_1, \ldots, N_d) \sim f_N$ independent of X and $f_N(x) = \frac{1}{C_N^d \sigma^d} e^{-\|x\|^p / (p\sigma^p)}$ Then*

$$D_{KL}(P_Y \| P_G * f_N) \leq \frac{1}{2\sigma^2} \left(W_{2,2\sigma^2}(P_Y, P_G) - W_{2,2\sigma^2}(P_Y, P_X) \right), \tag{15}$$

where $D_{KL}(P\|Q)$ is the KL-divergence ($D_{KL}(P\|Q) = \int P(x)\log\frac{P(x)}{Q(x)}dx$ for continuous P_X and $D_{KL}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}$ for discrete X)

Proof. By the formula for convolution: $P_G * f_N(x) = \int f_N(x-g)dP_G(g) = \mathbb{E}[f_N(x-G)]$. Note that Y is a continuous random variable, and plugging its density P_Y into the definition of KL-divergence we get:

$$\begin{aligned} D_{KL}(P_Y\|P_G * f_N) &= \int \log\frac{P_Y(y)}{P_G * f_N(y)}P_Y(y)dy \\ &= \mathbb{E}\log\frac{P_Y(Y)}{P_G * f_N(Y)} \\ &= \mathbb{E}\log P_Y(Y) - \mathbb{E}\left[\log\left(\mathbb{E}[f_N(Y-G)\mid Y]\right)\right] \\ &= -h(Y) - \mathbb{E}\left[\log\mathbb{E}[f_N(Y-G)\mid Y]\right]. \end{aligned} \tag{16}$$

The main ingredient for the rest of the proof will be the Donsker and Varadhan's variational formula (Donsker and Varadhan, 1983): for $U\sim P_U$ being a random variable supported on $\mathcal{U}$ and any measurable function $f:\mathcal{U}\to\mathbb{R}$, such that $\mathbb{E}[|f(U)|]<\infty$, it holds that

$$\log\mathbb{E}_{U\sim P_U}\left[e^{f(U)}\right] = \sup_{P_V\ll P_U}\left\{\mathbb{E}_{V\sim P_V}[f(V)] - D_{KL}(P_V\|P_U)\right\}, \tag{17}$$

where $P_V \ll P_U$ indicates that V is absolutely continuous with respect to U.

Now, we can use (17) to expand $\log\left(\mathbb{E}[f_N(Y-G)\mid Y=y]\right)$ in the negative term in (16). We fix some $y\in\mathbb{R}^d$ and choose $P_U := P_G$ and $f(g) := f_N(y-g)$, then

$$\begin{aligned} \log\mathbb{E}[f_N(Y-G)\mid Y=y] &= \log\mathbb{E}[f_N(y-G)] \\ &= \sup_{P_V^y\ll P_G}\left\{\mathbb{E}_{V\sim P_V^y}[f_N(y-V)] - D_{KL}(P_V^y\|P_G)\right\}, \end{aligned}$$

where we renamed P_V into P_V^y to emphasise its dependence on y. Plugging the above into (16) produces:

$$D_{KL}(P_Y\|P_G * f_N) = -h(P_Y) - \mathbb{E}\left[\sup_{P_V^y\ll P_G}\mathbb{E}_{V\sim P_V^y}[\log f_N(Y-V)\mid Y] - D_{KL}(P_V^y\|P_G)\right]$$

Denote now $\pi(v\mid y) = P_V^y(v)$ for any $v,y\in\mathbb{R}^d$ and notice that the supremum can be taken outside of the expectation since it is taken for each y independently, which leads to

$$\begin{aligned} D_{KL}(P_Y\|P_G * f_N) &= -h(P_Y) \\ &- \sup_{\{\pi(v|y)\ll P_G\,|\,y\in\mathbb{R}^d\}}\mathbb{E}\left[\mathbb{E}_{V\sim\pi(\cdot|Y)}[\log f_N(Y-V)\mid Y] - D_{KL}(\pi(\cdot\mid Y)\|P_G)\right]. \end{aligned} \tag{18}$$

Letting now $\mu(v) = \int\pi(v\mid y)P_Y(y)dy$, we get that $\pi(v,y) = \pi(v\mid y)P_Y(y)$ is a coupling between μ and P_Y, i.e. $\pi\in\Pi(\mu,P_Y)$. Note that $\mu\ll P_G \iff \pi(v\mid y)\ll P_G$ since $P_Y(y)>0\forall y$. Moreover, the supremum can be taken outside of the expectation since it is taken for each y independently, which leads to

$$\begin{aligned} &D_{KL}(P_Y\|P_G * f_N) \\ &\quad= -\sup_{\mu\ll P_G}\sup_{\pi\in\Pi(\mu,P_Y)}\mathbb{E}_{(V,Y)\sim\pi}\log f_N(Y-V) - \mathbb{E}D_{KL}(\pi(\cdot\mid Y)\|P_G) - h(Y) \\ &\quad= \inf_{\mu\ll P_G}\inf_{\pi\in\Pi(\mu,P_Y)}\mathbb{E}_{(V,Y)\sim\pi}[-\log f_N(Y-V)] + \mathbb{E}D_{KL}(\pi(\cdot\mid Y)\|P_G) - h(Y) \end{aligned} \tag{19}$$

As a final step we use the chain rule for KL-divergence: for any two joint distributions $Q^1\ll Q^2$ with marginals (Q_X^1,Q_Y^1) and (Q_X^2,Q_Y^2) correspondingly, it holds that

$$\begin{aligned} D_{KL}(Q^1\|Q^2) &= D_{KL}(Q_X^1\|Q_X^2) + \mathbb{E}_{X\sim Q_X^1}D_{KL}(Q^1(\cdot\mid X)\|Q^2(\cdot\mid X)) &(20)\\ &= D_{KL}(Q_Y^1\|Q_Y^2) + \mathbb{E}_{Y\sim Q_Y^1}D_{KL}(Q^1(\cdot\mid Y)\|Q^2(\cdot\mid Y)). &(21) \end{aligned}$$

Setting $Q^1 = \pi$ and $Q^2 = P_G \times P_Y$, we can rewrite the D_{KL} term in (19) using (21) as

$$\begin{aligned}\mathbb{E}_{Y\sim P_Y} D_{KL}(\pi(\cdot \mid Y)\|P_G) &= D_{KL}(\pi\|P_G \times P_Y) - D_{KL}(\pi_Y\|P_Y)\\ &= D_{KL}(\pi\|P_G \times P_Y)\\ &= \mathbb{E}_{G\sim\mu} D_{KL}(\pi(\cdot \mid G)\|P_Y) + D_{KL}(\mu\|P_G),\end{aligned} \tag{22}$$

where in the last equality we used (20). For the first term in (23) we use (20) again with $Q^1 = \pi$ and $Q^2 = \mu \times P_Y$, which results in

$$\mathbb{E} D_{KL}(\pi(\cdot \mid Y)\|P_G) = \mathbb{E}_{G\sim\mu} D_{KL}(\pi\|\mu \times P_Y) + D_{KL}(\mu\|P_G) \tag{23}$$

We finally note that $I_\pi(G, Y) = D_{KL}(\pi\|\mu \times P_Y)$ by the definition of mutual information. Plugging this and (23) into (19) gives

$$\begin{aligned}&D_{KL}(P_Y\|P_G * f_N) =\\ &\inf_{\mu\ll P_G} \inf_{\pi\in\Pi(\mu,P_G)} \mathbb{E}_{(V,Y)\sim\pi}[-\log f_N(Y - V)] + I_\pi(G, Y) + D_{KL}(\mu\|P_G) - h(Y).\end{aligned}$$

Letting $\mu = P_G$ gives us the upper bound:

$$\begin{aligned}&D_{KL}(P_Y\|P_G * f_N)\\ &\quad\le \inf_{\pi\in\Pi(P_G,P_Y)} \mathbb{E}_{(G,Y)\sim\pi}[-\log f_N(Y - G)] + I_\pi(G, Y) - h(Y)\end{aligned}$$

We can now plug in $f_N(x) = \frac{1}{C_N^d \sigma^d} e^{-\|x\|_p^p/\sigma^p p}$:

$$\begin{aligned}&D_{KL}(P_Y\|P_G * f_N)\\ &\quad\le \inf_{\pi\in\Pi(P_G,P_Y)} \mathbb{E}_{(G,Y)\sim\pi}\left[\frac{\|Y - G\|_P^P}{p\sigma^p}\right] + I_\pi(G, Y) + d\log(C_N\sigma) - h(Y)\\ &\quad= \frac{1}{p\sigma^p}\left(W_{p,p\sigma^p}(P_Y, P_G) + dp\sigma^p \log(C_N\sigma) - p\sigma^p h(Y)\right)\\ &\quad= \frac{1}{p\sigma^p}\left(W_{p,p\sigma^p}(P_Y, P_G) - W_{p,p\sigma^p}(P_Y, P_X)\right),\end{aligned} \tag{24}$$

where we used $W_{p,p\sigma^p}(P_Y, P_X) = p\sigma^p(h(Y) - d\log(C_N\sigma))$ from (14). □

We now prove Part (ii) of Theorem 1

To show (10) we first prove that

$$W_{p,p\sigma^p}(P_{G(Z)}, P_Y) - W_{p,p\sigma^p}(P_X, P_Y) \le W_p^p(P_{G(Z)}, P_X). \tag{25}$$

We fix a coupling $\pi \in \Pi(P_{G(Z)}, P_X)$ and let $(G(Z), X) \sim \pi$ and $Y = X + N$ where $N \sim f_N$ is independent of $(X, G(Z))$. Then

$$\mathbb{E}[\|G(Z) - Y\|_p^p] \le \mathbb{E}[\|G(Z) - X\|_p^p] + \mathbb{E}[\|X - Y\|_p^p], \tag{26}$$

where for $p = 1$ this is the triangle inequality and for $p = 2$:

$$\begin{aligned}\mathbb{E}[\|G(Z) - Y\|_2^2] &= \mathbb{E}[\|G(Z) - X\|_2^2] + 2\mathbb{E}[(G(Z) - X)^T(X - Y) + \mathbb{E}[\|X - Y\|_2^2]\\ &= \mathbb{E}[\|G(Z) - X\|_2^2] + 2\mathbb{E}[(G(Z) - X)^T N] + \mathbb{E}[\|X - Y\|_2^2].\end{aligned}$$

By the independence of N and $G(Z), X$:

$$\mathbb{E}[(G(Z) - X)^T N] = \mathbb{E}[(G(Z) - X)]^T \mathbb{E}[N] = 0$$

since $\mathbb{E}[Y-X]=\mathbb{E}[N]=0$. So, (26) holds for $p=2$ as well as for $p=1$. Also note that $G(Z)-X-Y$ forms a Markov chain, so the data processing inequality holds:

$$I(G(Z),Y)\leq I(X,Y).$$

Thus, for any $\pi\in\Pi(P_{G(Z)},P_X)$ and for $Y=X+N$ with N independent of $G(Z),X$:

$$\mathbb{E}[G(Z)-Y\|_p^p]+p\sigma^p I(G(Z),Y)\leq\mathbb{E}[\|G(Z)-X\|_p^p]+\mathbb{E}[\|X-Y\|_p^p]+p\sigma^p I(X,Y)$$

Note that the infimum of the LHS over all couplings $\pi'\in\Pi(P_{G(Z)},P_Y)$ is by definition (7) $W_{p,p\sigma^p}(P_{G(Z)},P_Y)$. So, for any $\pi\in\Pi(P_{G(Z)},P_X)$:

$$W_{p,p\sigma^p}(P_{G(Z)},P_Y)\leq\mathbb{E}_{(G(Z),X)\sim\pi}[\|G(Z)-X\|_p^p]+\mathbb{E}[\|X-Y\|_p^p]+p\sigma^p I_\pi(X,Y)$$

Now taking the infimum over $\pi\in\Pi(P_{G(Z)},P_X)$ on the RHS and recalling that

$$\inf_{\pi\in\Pi(P_{G(Z)},P_X)}\mathbb{E}_\pi[\|X-G(Z)\|_p^p]=W_p^p(P_{G(Z)},P_X),$$

leads to

$$W_{p,p\sigma^p}(P_{G(Z)},P_Y)\leq W_p^p(P_{G(Z)},P_X)+\mathbb{E}[\|X-Y\|_p^p]+p\sigma^p I_\pi(X,Y)$$

The only thing left to show is that

$$\mathbb{E}[\|X-Y\|_p^p]+p\sigma^p I_\pi(X,Y)=W_{p,p\sigma^p}(P_X,P_Y),$$

but this follows from our choice of $Y=X+N$ and part (i) of Theorem 1.

Second, we show that

$$W_{p,p\sigma^p}(P_{G(Z)},P_Y)-W_{p,p\sigma^p}(P_X,P_Y)\geq D_{KL}(P_{G(Z)+N},P_{X+N}).$$

It follows directly from Lemma 5.1 by setting $G=G(Z)$. Combining this with (25) results in

$$D_{KL}(P_{G(Z)+N},P_{X+N})\leq W_{p,p\sigma^p}(P_{G(Z)},P_Y)-W_{p,p\sigma^p}(P_X,P_Y)\leq W_p^p(P_{G(Z)},P_X)$$

Letting

$$G^*=\arg\min_{G\in\mathcal{G}}W_{p,p\sigma^p}(P_{G(Z)},P_Y),$$

and taking the minimum on both sides of

$$W_{p,p\sigma^p}(P_{G(Z)},P_Y)-W_{p,p\sigma^p}(P_X,P_Y)\leq W_p^p(P_{G(Z)},P_X)$$

leads to

$$D_{KL}(P_{G^*(Z)+N},P_{X+N})\leq W_{p,p\sigma^p}(P_{G^*(Z)},P_Y)-W_{p,p\sigma^p}(P_X,P_Y)\leq\min_{G\in\mathcal{G}}W_p^p(P_{G(Z)},P_X).$$

5.3. Proof of Theorem 2 and Corollary 3

Theorem 2 follows from the following theorem proved in (Reshetova et al., 2021, Theorem 6).

Theorem 3. *(Reshetova et al., 2021, Theorem 6)*

Let P_Z and P_Y be sub-Gaussian and the set of generators $\mathcal{G}$ consist of L-Lipschitz functions, i.e. $\|G(Z_1)-G(Z_2)\|\leq L\|Z_1-Z_2\|$ for any Z_1,Z_2 in the support of P_Z and let $\mathcal{G}$ satisfy (11)*. Then the generalization error for entropic GAN with $p=2$* (8) *can be both upper bounded as*

$$\mathbb{E}\big[W_{2,\lambda}^2(P_{G^n(Z)},P_Y)-W_{2,\lambda}^2(P_{G^*(Z)},P_Y)\big]\leq C_d\lambda n^{-1/2}\big(1+(2\tau^2/\lambda)^{\lceil 5d/4\rceil+3}\big)\tag{27}$$

with $\tau^2=\max\{L^2\sigma^2(Z),\sigma^2(Y)\}$.

We present the proof of Theorem 2 below:

Proof. In our case $\lambda = 2\sigma^2$ and $Y = X + N$ with $N \sim \mathcal{N}(0, \sigma^2 I)$, so $\sigma(Y) \leq \sigma(X) + \sigma(N)$, where $\sigma(N) = \sigma^2$. Thus, plugging it into the theorem we get

$$\mathbb{E}\big[W^2_{2,\lambda}(P_{G^n(Z)}, P_Y) - W^2_{2,\lambda}(P_{G^*(Z)}, P_Y)\big] \\ \leq C_d \sigma^2 n^{-1/2} \left(1 + \left(\frac{\max\{L\sigma(Z), \sigma(X) + \sigma\}}{\sigma}\right)^{2\lceil \frac{5d}{4} \rceil + 6}\right)$$

Letting $\tau = \max\{L\sigma(Z)/\sigma(X), 1\}$ we get $(\sigma(X) + \sigma)\tau \geq \max\{L\sigma(Z), \sigma(X) + \sigma\}$, which leads to

$$\mathbb{E}\big[W^2_{2,\lambda}(P_{G^n(Z)}, P_Y) - W^2_{2,\lambda}(P_{G^*(Z)}, P_Y)\big] \\ \leq C_d \sigma^2 n^{-1/2} \left(1 + \left(\tau^2 \left(1 + \sigma(X)/\sigma\right)^2\right)^{\lceil \frac{5d}{4} \rceil + 3}\right).$$

□

We can now prove Corollary 3.

Proof. Denoting f_N the pdf of $\mathcal{N}(0, \sigma^2 I)$ and plugging it into lemma 5.1 with $P_G = P_{G^n(Z)}$ leads to

$$D_{KL}(P_Y \| P_{G^n(Z)} * f_N) \leq \frac{1}{2\sigma^2} \left(W_{2,2\sigma^2}(P_Y, P_{G^n(Z)}) - W_{2,2\sigma^2}(P_Y, P_X)\right).$$

Taking the expectation of both sides and applying Theorem 2 proves the claim. □

5.4. Additional Deatils on Experiments

Data Privatization For the Laplace mechanism we project the data onto an ℓ_1 ball, add the Laplace noise with scale ϵ for ϵ-LDP and add it to the training data. For the Gaussian mechanism we project the data onto an ℓ_2 ball, add the Gaussian noise with variance σ^2 calculated from (4) for (ϵ, δ)-LDP and add it to the training data. Then we proceed with training the model

GAN training For training the Sinkhorn GAN we follow the work of (Genevay et al., 2018) by using Sinkhorn-Knopp algorithm (Flamary et al., 2021) to approximate the optimal transport plan π in (8) from the mini-batches of size b both for the generated and privatized training data. We use 100-dimensional Uniform $[0, 1]$ noise at the input to the generator ($\mathcal{P}_z = \text{Unif}[0, 1]^{100}$). The algorithm is stated here for completeness, where θ stands for the parameter of the Generator, i.e. $\mathcal{G} = \{G_\theta : \mathcal{Z} \to \mathcal{X} \mid \theta \in \Theta\}$

Algorithm 1 Training GAN with $W_{p,p\sigma^p}$

Input: θ_0, $\{y_i\}_{i=1}^n$ (the privatized training data), b (batch size), L (number of Sinkhorn iterations), σ (noise scale), α (learning rate), p (for ℓ_p-distance as the cost)
Output: θ
  $\theta \leftarrow \theta_0$
  for $t = 1, 2, \ldots$ **do**
    Sample $\{y_i\}_{i=1}^b$ from the train set, $Q_Y^b := \frac{1}{b}\sum_{i=1}^b \delta_{y_i}$
    Sample $\{z_i\}_{i=1}^b \overset{\text{i.i.d}}{\sim} P_{\mathcal{Z}}$, $Q_G^b := \frac{1}{b}\sum_{i=1}^b \delta_{G_\theta(z_i)}$
    Calculate $\pi = \arg\min_{\pi \in \Pi(Q_G^b, Q_Y^b)} \mathbb{E}_{(G_\theta(Z), Y) \sim \pi}[\|G_\theta(Z) - Y\|_p^p] + p\sigma^p I_\pi(G_\theta(Z), Y)$ – the optimal transport plan for $W_{p,p\sigma^p}(Q_G^b, Q_Y^b)$ with L Sinkhorn-Knopp steps
    $C_{ij} \leftarrow \|y_i - G_\theta(z_j)\|_p^p$ for $i, j = 1, \ldots b$
    $g_t \leftarrow \nabla_\theta \langle \pi, C \rangle$
    $\theta \leftarrow \theta - \alpha g_t$.
  end for

Dataset and architecture We train our models on MNIST data (LeCun, 1998), consisting of 60000 grayscale images of handwritten digits, we do not use the labels to mimic a fully unsupervised training scenario. The generator model is DCGAN from (Radford et al., 2015) with latent space dimension 100. All the losses were used in the primal formulations (5),(7) with optimization over the coupling matrix.
Details on section 4 For training the entropic p-WGAN we use the 400 Sinkhorn-Knopp iterations and the Adam optimizer with learning rate 10^{-4} for 100 epochs. No clipping of the norm of images was performed. In Figure 2 we provide 200 uniformly random samples for Entropic p-WGAN on MNIST trained on data privatized with the corresponding mechanism together with the privacy parameter.

5.5. MNIST: Higher Privacy samples

In this experiment we illustrate the results we got with the entropic p-Wasserstein GAN with LDP. We set the number of sinkhorn steps $L = 400$ and the batchsize to be $b = 400$ and we performed optimization with Adam optimizer (Kingma and Ba, 2014) and learning rate varied in $\{0.005, 10^{-4}, 5 \times 10^{-5}\}$. We optimized for 150 epochs. For $p = 1$ we first took the discrete cosine transform of the images and clipped the coefficients below 0.8 quantile to preserve more information. We also applied DCT transform to the generator output before plugging it into the loss function.

The results indicate the effectiveness of our model at higher privacy regimes, however, smaller ϵ values still produced a lot of noise in the generated samples or eroded the images significantly. This can be potentially mitigated by increasing the number of samples as suggested in Theorem 2, however the relatively small size of MNIST limits the privacy levels that can be achieved.

We additionally report 400 randomly sampled digits with different privacy levels in figure 5 for the Laplace mechanism and in figure 6 for the Gaussian mechanism.

We discuss convergence in more detail in the next section.

5.6. Empirical convergence

Third, we empirically check how the performance (measured by the 2-Wasserstein distance) depends on the privatization level. In our experiment, we set $p = 2$ and train GANs with 3 different loss functions on MNIST: the entropy-regularized 2-Wasserstein loss between the generated distribution and the empirical distribution of the privatized samples $W_{2,2\sigma^2}(P_{G(Z)}, Q_Y^n)$, the 2-Wasserstein distance $W_2^2(P_{G(Z)}, Q_Y^n)$ and the sinkhorn divergence (Feydy et al., 2019) (which is the debiased version of the entropy-regularized 2-Wasserstein distance). We choose the latent dimension to be 2-dimensional, i.e. $P_Z = \text{Uniform}[0,1]^2, L = 200, b = 200$. We report the 2-Wasserstein distance between the generated and the target distribution P_X, on Figure 7 where we approximate the target distribution by the empirical distribution of the non-privatized data. The results show that the distance grows with the noise scale σ for all three of the metrics we considered, however, the slope of our method $W_{2,2\sigma^2}(P_{G(Z)}, Q_Y^n)$ is the smallest. The growth is to be expected from theorem 2, when the dataset size n is kept constant, increasing the noise scale σ (and thus privatization) degrades the performance.

5.7. Influence of train set size on the error

Here we empirically check how the performance (measured by the 2-Wasserstein distance) depends on the size of the training set . We use the setting described in section 5.6 with $\sigma^2 = 4$ and train the model with the entropy-regularized 2-Wasserstein loss between the generated distribution and the empirical distribution of the privatized samples $W_{2,2\sigma^2}(P_{G(Z)}, Q_Y^n)$. We report the 2-Wasserstein distance between the generated $P_{G(Z)}$ and the target distribution P_X on Figure 8, where we approximate the target distribution by the empirical distribution of the non-privatized data that was used for training (curve labeled "train") and that was left out for validation (curve labeled "test"). To compute $W_{2,2\sigma^2}(P_{G(Z)}, P_Y)$ we use mini-batches of size 200. The distance is decreasing on the left plot for both train and test curves, which is expected by theorem 2 to be proportional to $1/\sqrt{n}$. The closeness of the train and test curves also shows no signs of overfitting, which is most probably happening due to privatization.

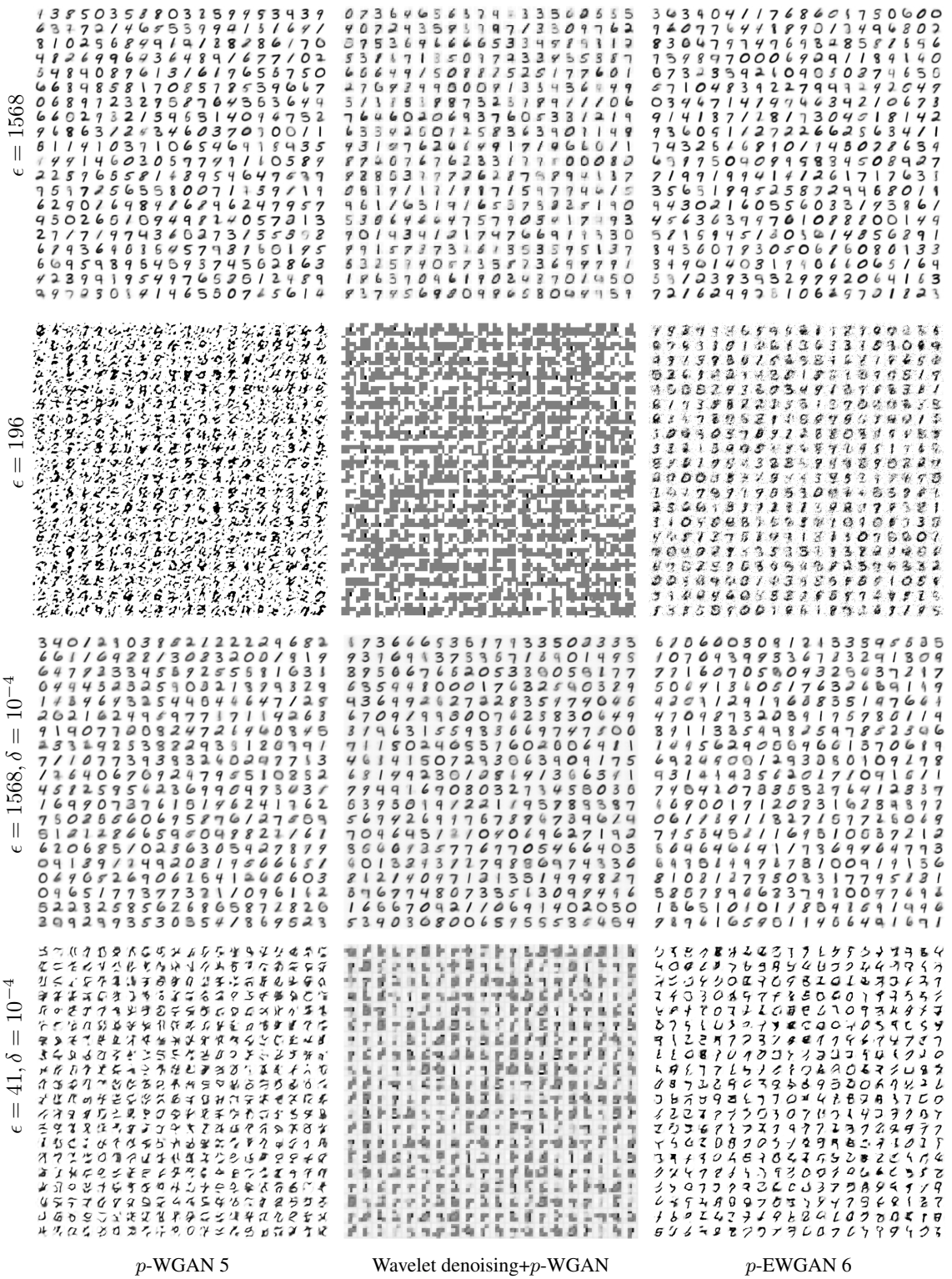
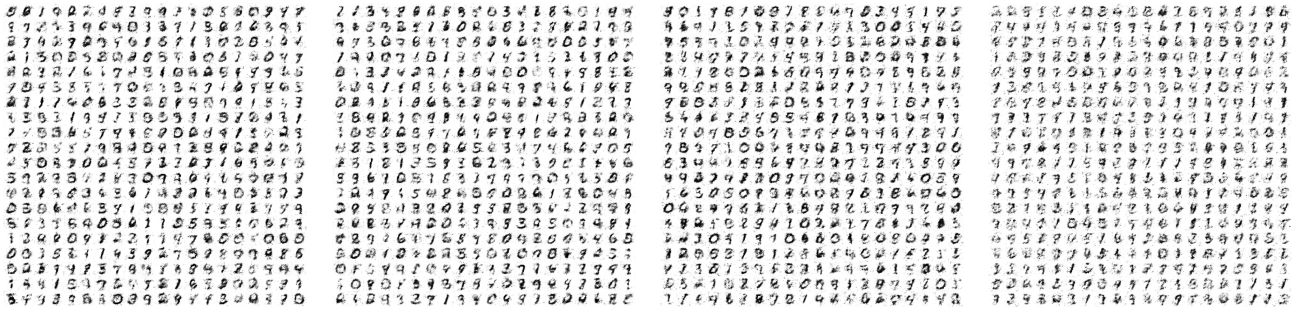


Figure 2: Comparing Wavelet denoising and Entropic p-WGAN for $p = 1$ and Laplace mechanism (top 2 rows) and $p = 2$ and Gaussian mechanism (bottom 2 rows)

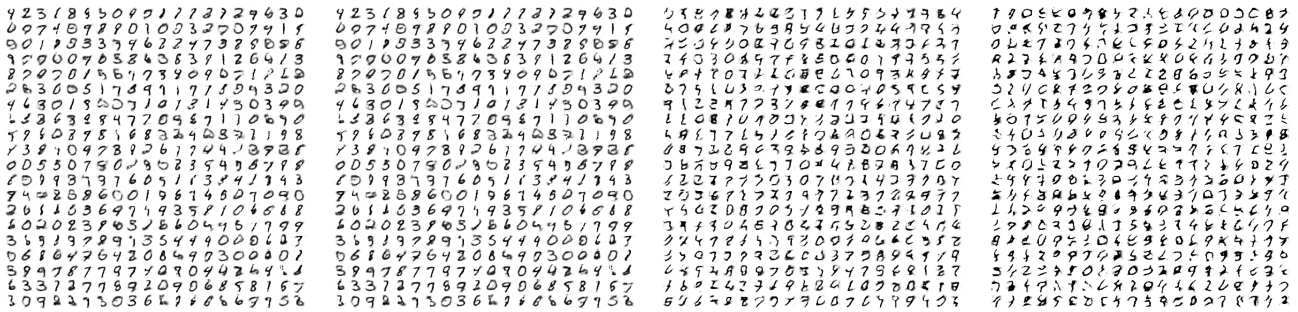
Figure 3: Entropic 1-WGAN on MNIST trained on data privatized with the Laplace mechanism achieving ϵ-LDP $\epsilon = 35$ (left) and $\epsilon = 25$ (right)

Figure 4: Entropic 2-WGAN on MNIST trained on data privatized with the Gaussian mechanism achieving (ϵ, δ)-LDP with $\delta = 10^{-4}$ and $\epsilon = 30$ (left) and $\epsilon = 25$ (right)



(a) $\epsilon = 65.25$ (b) $\epsilon = 49$ (c) $\epsilon = 35$ (d) $\epsilon = 25$

Figure 5: Laplace mechanism with different privacy budgets ϵ and clipping of the discrete cosine transform, 1-EWGAN



(a) $\epsilon = 35$ (b) $\epsilon = 30$ (c) $\epsilon = 25$ (d) $\epsilon = 25$

Figure 6: Gaussian mechanism with different privacy budgets ϵ and clipping the euclidean norm of images, 2-EWGAN

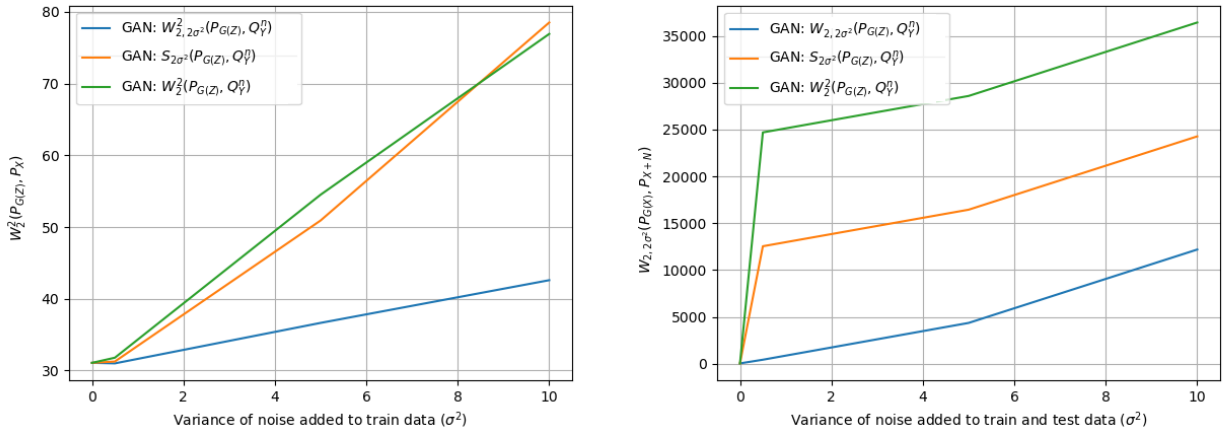


Figure 7: Dependence of the 2-Wasserstein distance(left) and the validation error (right) on the noise scale σ

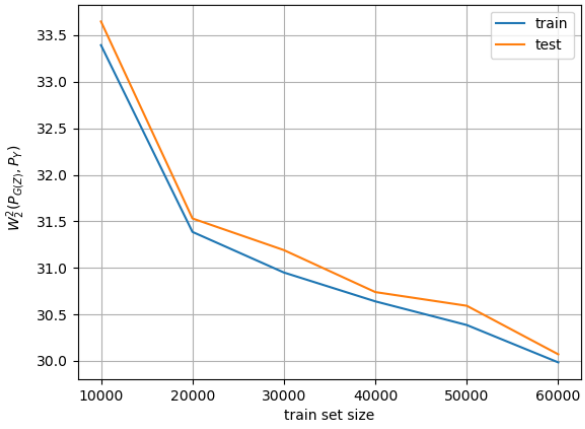


Figure 8: Dependence of $W_{2,2\sigma^2}(P_{G(Z)}, P_Y)$ on the dataset size n