Principled Private Data Release with Deep Learning

Michael Fine Harvard University
Cambridge MA, 02138
mfine@college.harvard.edu

November 30, 2019

1 Introduction

2 Related Work

- [JY19]
- [GLL+17]
- [NRVW19]
- [AGH18]
- [NRW18]
- [GAH+14]
- [HLM12]
- [GXC+18]

3 Background

3.1 Differential Privacy

3.2 Query Release Problem

We study the problem of privately generating synthetic data to answer statistical queries over a data universe $\mathcal{X}$. Formally, a statistical query over $\mathcal{X}$ is a function $q : \mathcal{X} \to \{0, 1\}$. Given a a dataset $x \in \mathcal{X}^n$, we define $q(x) := \sum_{i=0}^{n} q(x_i)$. For convenience, we will often normalize queries to take values $\in [0, 1]$

$$q(x) := \frac{1}{n} \sum_{i=0}^{n} q(x_i) = \mathbb{E}_{x_i \in \mathcal{X}} q(x_i) \tag{1}$$

Our goal is to produce a synthetic dataset that, for every query in some family of queries, takes approximately the same value as the true dataset.

Definition 3.1 ($\alpha-$approximate)**.** *We say a synthetic dataset x $\alpha-$approximates a true dataset $\hat{x}$ w.r.t a family of statistical queries $\mathcal{Q}$ if*

$$\forall q \in \mathcal{Q} : \quad |q(x) - q(\hat{x})| \leqslant \alpha \tag{2}$$

3.3 Game Theoretic Formulation

One can formulate the problem of producing an $\alpha-$approximate dataset as a two-player, zero sum game [HRU13] between a discriminator D and a generator G. The generator has an action set $\mathcal{X}$, while the discriminator has an action set $\mathcal{Q}$. The generator aims to output a dataset $x \in \mathcal{X}$ that maximally agrees with $\hat{x}$, while the discriminator aims to find queries $q \in \mathcal{Q}$ that distinguish $\hat{x}$ and x.

Formally, given a play $x \in \mathcal{X}$ and $q \in Q$, the discriminator gets payoff $V(x, q)$ and the generator gets payoff $-V(x, q)$, where $V(x, q)$ denotes:

Definition 3.2 (Payoff)**.**

$$V(x, q) := q(x) - q(\hat{x}) \tag{3}$$

The goal of both G and D is to maximize their worst case payoffs, thus

$$\max_{q \in \mathcal{Q}} \min_{x \in X} V(x, q) \quad \text{(Goal of } D\text{)} \qquad \textit{and} \qquad \min_{x \in X} \max_{q \in \mathcal{Q}} V(x, q) \quad \text{(Goal of } G\text{)} \tag{4}$$

If there exists a point (x^*, q^*) such that neither G nor D can improve their payoffs by playing a different move, we call that a *Pure Nash Equilibrium*. Unfortunately, a pure equilibrium is not always guaranteed to exist (and likely does not in the case of synthetic data generation).

However, the seminal work of Nash et. al showed that there always exists a *Mixed Nash Equilibrium (MNE)*, where the players play *probability distributions* over their action sets, instead of fixed actions.

Let $\Delta(\mathcal{X})$ and $\Delta(\mathcal{Q})$ denote the set of probability distribution over $\mathcal{X}$ and $\mathcal{Q}$. Formally, if G plays a strategy $g \in \Delta(\mathcal{X})$ and D plays $d \in \Delta(\mathcal{Q})$, we define the payoff to be the expected value of a single draw:

$$V(g, d) := \mathbb{E}_{x \sim g, q \sim d} V(x, q) \tag{5}$$

Thus, a pair of strategies $g \in \Delta(\mathcal{X})$ and $d \in \Delta(\mathcal{Q})$ forms an $\alpha-$approximate mixed nash equilibrium if for all strategies $u \in \Delta(\mathcal{X})$ and $w \in \Delta(\mathcal{Q})$

$$V(g, w) \leqslant V(u, w) + \alpha \qquad \textit{and} \qquad V(u, d) \leqslant V(u, w) - \alpha \tag{6}$$

Moreover, Gaboardi et. al showed how to reduce the problem of finding an $\alpha-$approximate dataset to the problem of finding an an $\alpha-$equilibrium in the query release game:

Theorem 3.1. *Let (u, w) be the $\alpha-$approximate MNE in a query release game for a dataset $\hat{x} \in \mathcal{X}$ and a query universe $\mathcal{Q}$. If $\mathcal{Q}$ is closed under negation, then the dataset S sampled from u α-approximates $\hat{x}$ over Q. [GAH+14]*

Hence, our task is to provide an algorithm to private reach an $\alpha-$MNE in the query release game. In the following section, we will provide the background for how this can be done with GANs.

Heuristics paper summarized this better and more concisely

3.4 Online Learning

To efficiently find equilibrium in the zero-sum GAN game, we draw on results from online learning. In the online learning setting, in each of T rounds a player is given a loss function f_t , possibly adversarial chosen. The players goal is to chose an action $x_{t+1} \in \mathcal{X}$ in order to minimize the cumulative *regret*.

Definition 3.3 (Regret)**.** *The regret measures the cumulative loss of the player, compared to the best fixed decision in hindsight.*

$$Regret_T(f_1, ..., f_T) = \sum_{t=1}^{T} f_t(x_t) - min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x^*) \tag{7}$$

When a strategy provably leads to regret is sublinear in T, we call that *no-regret*, as regret $\rightarrow 0$ as $T \rightarrow \infty$. One approach to regret minimization is to choose the action x_{t+1} that minimizes the cumulative loss over all past loss functions l

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) \tag{8}$$

This approach is known as *Follow-The-Leader*. While natural, this approach is easily exploitable by an adversary. At a high level, this is because it *overfits* to past outcomes, allowing it to optimize between suboptimal strategies. To rectify this, a powerful strategy is *Follow-the-Regularized-Leader*

Definition 3.4 (Follow-The-Reguralized-Leader (FTRL)). *Given a reguralization function $R(x)$ and a regularization weight η_T, at each step choose $x_{t+1} \in \mathcal{X}$ to minimize the regularized cumulative loss:*

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} + \eta_T R(X) \tag{9}$$

One common regularization function is the l_2 norm $R(x) = ||x||_2$.

When the loss function f_T is convex, FTRL can be shown to be no-regret [Haz19]. Unfortunately, in the GAN setting, where the loss function f_T is defined by a highly non-convex deep network, we don't have that guarantee.

3.4.1 Online Learning for Non-convex losses

In general, finding the minimum of a sum of non-convex functions is hard. However, in practice gradient descent over neural networks has proven to be remarkably effective at approximately solving non-convex loss functions. This leads to the natural question: assuming we have an offline non-convex optimization oracle $\mathcal{O}$, can we use that to find a no-regret strategy in the online setting? Agarwal et al showed that we can, using a Follow-The-Leader variant known as Follow-The-Perturbed-Leader [AGH18]. Formally:

Definition 3.5 (Offline optimization oracle). *Let $\mathcal{O}$ take a sequence of (possibly non-convex) loss functions $(f_1 ... f_T) \in \mathcal{L}^T$ and a d-dimensional vector d, and output $x^* \in \mathcal{X}$*

$$x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) + \sigma^T x \tag{10}$$

If we relax the requirement to allow $\mathcal{O}$ to output an approximate minimizer x^*

$$x^* \leqslant \arg\min_{x \in \mathcal{X}} \left(\sum_{t=1}^{T} f_t(x) + \sigma^T x \right) + \alpha \tag{11}$$

we then call $\mathcal{O}$ an $\alpha-$approximate offline oracle.

It's not that $x^* \leqslant \alpha$, it's that $f(x^*)$ should be within α

We can use this offline oracle $\mathcal{O}$ to minimize regret in the online case:

Definition 3.6 (Follow-The-Perturbed-Leader (FTPL)). *Given an offline oracle $\mathcal{O}$ and a parameter η, at each step t FTPL draws a random vector $\sigma_t \sim Exp(\eta)^d$. It then outputs*

$$x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) + \sigma^T x \tag{12}$$

Theorem 3.2. *FTPL has sublinear regret. [AGH18]*

Building on the work of Freund and Schapire, we can show that if G and D both play FTPL for T rounds, they will converge to an $\alpha-$approximate equilibrium. Formally:

Theorem 3.3. *Suppose that G and D play according to FTPL. We can choose $T \in poly(d)/\alpha^3$ such that the expected average regret of FTPL is at most α. Then, G_T and D_T produce strategies in an $\alpha-$approximate equilibrium [AGH18].*

4 Differentially Private FTPL

Remember, our goal is to find a way to privately produce a synthetic dataset. To do so under differential privacy requires adding noise in the computation. Interestingly, FTPL already adds noise drawn from the exponential distribution to the loss function at each step. This is not to ensure privacy, but rather convergence of the online algorithm. .]

Move this to make clear this is new research

However, the assymetry of noise drawn from the exponential distribution makes it insufficient to ensure privacy. Consider the most basic mechanism that reports an average:

Deep connections between deep learning and differential privacy

$$M(x) = \sum_{i}^{n} x_i + Exp(\lambda) \tag{13}$$

Recall that to satisfy differential privacy, for all neighboring databases x, x',

$$\mathbb{P}(M(x) = B) \leqslant e^{\epsilon}\mathbb{P}(M(x') = B) \tag{14}$$

But consider a database x s.t $\sum_i^n x_i = 3$, and a neighboring database x' where $\sum_i^n x'_i = 1$. When $B = 2$

$$\frac{\mathbb{P}(M(x) = B)}{\mathbb{P}(M(x') = B)} = \frac{f(B - \sum_i^n x_i; \lambda)}{f(B - \sum_i^n x'_i; \lambda)} \tag{15}$$

$$= \frac{f(-1; \lambda)}{f(1; \lambda)} \tag{16}$$

$$= \frac{0}{\lambda e^{-\lambda}} \tag{17}$$

where $f(x; \lambda)$ is the exponential pdf.

Clearly, there is no value of ϵ or λ s.t. $\frac{0}{\lambda e^{-\lambda}} \geqslant e^{\epsilon}$, and therefore the exponential mechanism is not ϵ-DP for any value of epsilon.

4.1 Symmetric FTPL

Instead, to ensure privacy of FTPL, we must instantiate FTPL with a symmetric distribution. One natural choice is Laplace noise, but Gaussian noise is equally plausible. We will prove accuracy, and privacy then, for a number of symmetric noise distributions

Algorithm 1: Follow the Symmetric Perturbed Leader (FTSPL)

Input: Noise distribution $p_z \in \mathbb{R}^d$, Rounds T, Loss Distribution $L(f_{1:t})$
Result: Actions $x_{1:T}$
for $t \in 1...T$ **do**
  Draw i.i.d random vector $\sigma \sim p_z$
  Prediction at time t:

$$x_t \leftarrow \arg\min_{x \in \mathcal{X}} \sum_{f \in B} f(x) - \sigma^T x$$

We will now prove regret bounds for FTSPL. This proof draws heavily from [SN19], which used a stability argument to prove the following regret bound for non-convex FTPL with an (assymetric) exponential distribution:

Lemma 4.1 (Exponential Non-convex FTPL [SN19])**.** *Let D be the l_∞ diameter of the data universe $\mathcal{X}$. Suppose the losses $(f_1..f_t)$ are $L-lipschitz$. If FTPL is instantiated with noise distribution $Exp(\eta)$ and an $(\alpha, \beta)-$approximate optimization oracle $\mathcal{O}$, then FTPL satisfies the following regret bound*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \frac{1}{T}\min_{x \in \mathcal{X}}\sum_{t=1}^{T} f_t(x)\right] \leqslant O(\eta d^2 D L^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \beta d L) \tag{18}$$

Theorem 4.1 (FTSPL Regret)**.** *Let D be the l_∞ diameter of the data universe $\mathcal{X}$. Suppose the losses $(f_1..f_t)$ are $L-$lipschitz. If FTPL is instantiated with noise distribution p_z and an $(\alpha, \beta)-$approximate optimization oracle $\mathcal{O}$, then FTSPL satisfies the following regret bound*

$$TOOD \tag{19}$$

Our proof follows the exact same form, only substituting in the PDF of our symmetric distribution p_z for the exponential PDF where appropriate. We rely on the following two monotocity lemmas from , shown without proof as neither relies on the noise PDF, and thus are equivalent to those of [SN19].

First, let e_i denote the i^{th} standard basis vector and $x_{t,i}$ denote the i^{th} cooridinate of x_t.

Lemma 4.2 (Monotocity 1 [SN19])**.** *Let $x_t(\sigma)$ be the prediction of FTSPL in iteration t with random perturbation σ. Then, for any $c > 0$, the following monotonicity property holds*

$$\mathbf{x}_{t,i}\left(\sigma + c\mathbf{e}_i\right) \geqslant \mathbf{x}_{t,i}(\sigma) - \frac{2\left(\alpha + \beta\|\sigma\|_1\right)}{c} - \beta \tag{20}$$

Lemma 4.3 (Monotonicity 2)**.** *Let $x_t(\sigma)$ be the prediction of FTSPL in iteration t with random perturbation σ. Suppose $\|\mathbf{x}_t(\sigma) - \mathbf{x}_{t+1}(\sigma)\|_1 \leqslant 10d \cdot |\mathbf{x}_{t,i}(\sigma) - \mathbf{x}_{t+1,i}(\sigma)|$. For $\sigma' = \sigma + 100Lde_i$, we have*

$$\begin{aligned}\min\left(\mathbf{x}_{t,i}\left(\sigma'\right), \mathbf{x}_{t+1,i}\left(\sigma'\right)\right) \geqslant &\max\left(\mathbf{x}_{t,i}(\sigma), \mathbf{x}_{t+1,i}(\sigma)\right) - \frac{1}{10}\left|\mathbf{x}_{t,i}(\sigma) - \mathbf{x}_{t+1,i}(\sigma)\right| \\ &- \frac{3\left(\alpha + \beta\|\sigma\|_1\right)}{100Ld} - \beta\end{aligned}$$

Using these lemmas, we present our modified proof of 4.1

Proof of Theorem ??. We now proceed to the proof of Theorem **??**. We use the same notation as in Lemmas **??**, **??**. First note that $\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]$ can be written as

$$\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right] = \sum_{i=1}^{d} \mathbb{E}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right]. \tag{21}$$

To bound $\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]$ we derive an upper bound for $\mathbb{E}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right], \forall i \in [d]$. For any $i \in [d]$, define $\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right]$ as

$$\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] := \mathbb{E}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)| \Big| \{\sigma_j\}_{j\neq i}\right],$$

where σ_j is the j^{th} coordinate of σ. Let $\vec{x}_{max,i}(\sigma) = \max\left(\vec{x}_{t,i}(\sigma), \vec{x}_{t+1,i}(\sigma)\right)$ and $\vec{x}_{min,i}(\sigma) = \min\left(\vec{x}_{t,i}(\sigma), \vec{x}_{t+1,i}(\sigma)\right)$. Then $\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] = \mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - \mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right]$. Define event $\mathcal{E}$ as

$$\mathcal{E} = \left\{\sigma : \|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1 \leqslant 10d \cdot |\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right\}.$$

Consider the following

$$\begin{aligned}\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right] &= \Pr(\sigma_i < 100Ld)\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)|\sigma_i < 100Ld\right] \\ &\quad + \Pr(\sigma_i \geqslant 100Ld)\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)|\sigma_i \geqslant 100Ld\right] \\ &\geqslant \left(1 - f_z(100Ld)\right)\left(\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D\right) \\ &\quad + f_z(100Ld)\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma + 100Lde_i)\right],\end{aligned}$$

where the last inequality follows the fact that the domain of i^{th} coordinate lies within some interval of length D and since $\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)|\sigma_i < 100Ld\right]$ and $\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right]$ are points in this interval, their difference is bounded by D. We can further lower bound $\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right]$ as follows

$$\begin{aligned}\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right] &\geqslant \left(1 - f_z(100Ld)\right)\left(\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D\right) \\ &\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E})\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma + 100Lde_i)|\mathcal{E}\right] \\ &\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E}^c)\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma + 100Lde_i)|\mathcal{E}^c\right],\end{aligned}$$

where $\mathbb{P}_{-i}(\mathcal{E})$ is defined as $\Pr_{-i}(\mathcal{E}) := \Pr\left(\mathcal{E}\middle|\{\sigma_j\}_{j\neq i}\right)$. We now use the monotonicity properties proved in Lemmas **??**, **??** to further lower bound $\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right]$. Let $\gamma(\sigma) = \alpha + \beta\|\sigma\|_1$ be the approximation error of the offline optimization oracle. Then

$$\begin{aligned}
\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right] \quad &\geqslant \quad (1 - f_z(100Ld))\left(\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D\right) \\
&\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E})\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma) - \tfrac{1}{10}|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)| - \tfrac{3\gamma(\sigma)}{100Ld} - \beta\middle|\mathcal{E}\right] \\
&\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E}^c)\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma) - \tfrac{2\gamma(\sigma)}{100Ld} - \beta|\mathcal{E}^c\right] \\
&\geqslant \quad (1 - f_z(100Ld))\left(\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D\right) \\
&\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E})\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma) - \tfrac{1}{10}|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)| - \tfrac{3\gamma(\sigma)}{100Ld} - \beta\middle|\mathcal{E}\right] \\
&\quad + f_z(100Ld)\Pr_{-i}(\mathcal{E}^c)\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma) - \tfrac{1}{10d}\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1 - \tfrac{2\gamma(\sigma)}{100Ld} - \beta\middle|\mathcal{E}^c\right],
\end{aligned}$$

where the first inequality follows from Lemmas **??**, **??**, the second inequality follows from the definition of $\mathcal{E}^c$. Rearranging the terms in the RHS and using $\Pr_{-i}(\mathcal{E}) \leqslant 1$ gives us

$$\begin{aligned}
\mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right] \quad &\geqslant \quad (1 - f_z(100Ld))\left(\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D\right) \\
&\quad + f_z(100Ld)\mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma) - \tfrac{3\gamma(\sigma)}{100Ld} - \beta\right] \\
&\quad - f_z(100Ld)\mathbb{E}_{-1}\left[\tfrac{1}{10}|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)| + \tfrac{1}{10d}\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right] \\
&\geqslant \quad \mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - D - f_z(100Ld)\left(\tfrac{3\gamma(\sigma)}{100Ld} + \beta + D\right) \\
&\quad - f_z(100Ld)\mathbb{E}_{-1}\left[\tfrac{1}{10}|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)| + \tfrac{1}{10d}\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]
\end{aligned}$$

where the last inequality uses the fact that f_z is always less than 1 . Rearanging the terms in the last inequality gives us

$$\begin{aligned}
\tfrac{f_z(100Ld)}{10}\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] \quad &\leqslant \quad \mathbb{E}_{-1}\left[\vec{x}_{max,i}(\sigma)\right] - \mathbb{E}_{-1}\left[\vec{x}_{min,i}(\sigma)\right] - D \\
&\quad - f_z(100Ld)\left(\mathbb{E}_{-1}\left[\tfrac{3\gamma(\sigma)}{100Ld}\right] + \beta + D + \tfrac{1}{10d}\mathbb{E}_{-1}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]\right) \\
\tfrac{f_z(100Ld)-10}{10}\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] \quad &\leqslant \quad -D - f_z(100Ld)\left(\mathbb{E}_{-1}\left[\tfrac{3\gamma(\sigma)}{100Ld}\right] + \beta + D + \tfrac{1}{10d}\mathbb{E}_{-1}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]\right) \\
\mathbb{E}_{-1}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] \quad &\geqslant \quad \xi\left(\mathbb{E}_{-1}\left[\tfrac{3\gamma(\sigma)}{10Ld}\right] + 10\beta + 10D + \tfrac{D}{f_z(100Ld)} + \tfrac{1}{d}\mathbb{E}_{-1}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]\right)
\end{aligned}$$

where $\xi = \frac{f_z(100Ld)}{10 - f_z(100Ld)}$.Since the above bound holds for any $\{\sigma_j\}_{j\neq i}$, we get the following bound on the unconditioned expectation

$$\mathbb{E}\left[|\vec{x}_{t,i}(\sigma) - \vec{x}_{t+1,i}(\sigma)|\right] \leqslant \xi\left(\mathbb{E}\left[\frac{3\gamma(\sigma)}{10Ld}\right] + 10\beta + 10D + \frac{D}{f_z(100Ld)} + \frac{1}{d}\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]\right)$$

Plugging this in Equation (21) gives us the following bound on stability of predictions of FTPL

$$\begin{aligned}
\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right] &\leqslant d \cdot \xi\left(\mathbb{E}\left[\frac{3\gamma(\sigma)}{10Ld}\right] + 10\beta + 10D + \frac{D}{f_z(100Ld)} + \frac{1}{d}\mathbb{E}\left[\|\vec{x}_t(\sigma) - \vec{x}_{t+1}(\sigma)\|_1\right]\right) \\
&\leqslant d \cdot \frac{\xi}{\xi + 1}\left(\mathbb{E}\left[\frac{3\gamma(\sigma)}{10Ld}\right] + 10\beta + 10D + \frac{D}{f_z(100Ld)}\right) \\
&\leqslant f_z(100Ld)\left(\frac{3\alpha + 3\beta\mathbb{E}\left[\|\sigma\|_1\right]}{100L} + \beta d + Dd\right) + Dd/10
\end{aligned}$$

Plugging the above bound in Equation (**??**) gives us the following bound on regret.

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \frac{1}{T}\min_{x\in\mathcal{X}}\sum_{t=1}^{T} f_t(x)\right] \leqslant \frac{1}{T}[T(f_z(100Ld)(\frac{3\alpha+3\beta\mathbb{E}\left[\|\sigma\|_1\right]}{100L} + \beta d + Dd) + Dd/10) + TODOtheotherterms] \tag{22}$$

$$\leqslant O\left((f_z(100Ld)(\frac{\alpha+\beta\mathbb{E}\left[\|\sigma\|_1\right]}{L} + \beta d + Dd) + Dd) + TODOtheotherterms\right) \tag{23}$$

4.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Goodfellow et. al, are an approach to generative deep learning that has shown remarkable promise in generating high fidelity samples [GPM+14]. In the GAN setup, a generator G is paired with a discriminator D. At each round, D is trained to distinguish real samples drawn from P_{data} from generated samples drawn from P_g, while G is trained to generate realistic samples that fool the discriminator.

This yields a two player, zero sum game with minimax objective

$$\min_G \max_D V_{gan}(G,D) := \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}} \log D(\mathbf{x}) + \frac{1}{2}\mathbb{E}_{z\sim p_z}\log(1-D(G(z))) \tag{24}$$

However, [ACB17] showed that this cost function V_{gan} is not sensible cost function in practice, when the distributions are supported by low-dimensional manifolds. Instead, they proposed to use the Earth-Mover, or Wasserstein-1 objective.

Definition 4.1 (Earth Mover Distance)**.** *The EM distance between two distribution $\mathbb{P}_r$ and $\mathbb{P}_g$ is*

$$W(\mathbb{P}_r,\mathbb{P}_g) := \inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}|x-y| \tag{25}$$

where $\Pi(\mathbb{P}_r,\mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x,y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$.

While this infinimum is highly intractable to compute, [ACB17] used the Kantorovich-Rubinstein duality [Vil08] to show that

$$W(\mathbb{P}_r,\mathbb{P}_g) = \sup_{\|f\|_L\leqslant 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)] \tag{26}$$

where the supremum is over all 1-Lipschitz functions.

Definition 4.2 (Lipschitz function)**.** *A function f is said to be $L-Lipschitz$ if*

$$|f(x)-f(y)| \leqslant C|x-y| \tag{27}$$

for all x,y in the domain

Thus, if our discriminators are parametrized by a family of 1-Lipschitz functions $\mathcal{D}$, the Wasserstein GAN objective is

$$\min_{G\in\mathcal{G}} \max_{D\in\mathcal{D}} \mathbb{E}_{\mathbf{x}\sim p_{data}}[D(\mathbf{x})] - \mathbb{E}_{z\sim p_z}[D(G(z))] \tag{28}$$

Remark 4.1. In the Wasserstein GAN, the discriminator is no longer guaranteed to output values in $[0,1]$, and therefore cannot be interpreted as a probability. For this reason, the Wasserstein discriminator is typically called a *critic.* .

Elaborate, maybe cite

Note the remarkable similarity of the GAN objective V_{gan} to the earlier query release objective V. Recall that the mixed strategy V is defined as [1]

$$\min_{\mathbb{P}_g \in \nabla \mathcal{X}} \max_{\mathbb{P}_q \in \nabla \mathcal{Q}} V(g,d) = |\mathbb{E}_{x \sim \mathbb{P}_g}[q(x)] - \mathbb{E}_{x \sim \mathbb{P}_{data}}[q(x)]|$$

Assuming that the class of queries $\mathcal{Q}$ is closed under negation, the absolute value can be dropped, and we recover the exact Wasserstein formulation. Thus, achieving equilibrium in the Wasserstein GAN is equivalent to solving the query release problem, for all queries representable by the discriminator. Importantly, for the Wasserstein GAN this limits us to all queries that are 1-Lipschitz over each row of the output. In this context, that may prove to be infeasibly limiting. .

Explain why this works for images but not for queries

Segue

4.2.1 Integral Probability Metrics

Many GAN variants can be understood as minimizing the distance between the two distributions $\mathbb{P}_{synth}$ and $\mathbb{P}_{real}$, measured by some Integral Probability Metric.

Definition 4.3 (Integral Probability Metric (IPM) [Mül97])**.** *An IPM $\rho_{\mathcal{F}}$ between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is*

$$\rho_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{x \sim \mathbb{P}}[f(X)] - \mathbb{E}_{x \sim \mathbb{Q}}[f(X)] \right| \tag{29}$$

where $\mathcal{F}$ is some class of real-valued bounded measurable functions.

The boundedness criteria is especially important, for if f is unbounded the objective $\sup_f$ will scale f to be arbitrarily large. Depending on how we constrain $\mathcal{F}$, we can recover a number of GAN architectures [zot]:

- $\mathcal{F} = \{f : \|f\|_L \leqslant 1\}$ gives us Wasserstein GAN [ACB17]
- $\mathcal{F} = \{f : \|f\|_\infty \leqslant 1\}$ gives us Total Variation distance, as seen is Energy Based GAN [ZML17]
- $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leqslant 1\}$ for some RKHS $\mathcal{H}$ gives us Maximum Mean Discrepancy, as seen in GMMN [LSZ15]

In practice, most GAN literature approximates these function class $\mathcal{F}$ through deep neural networks, rather than optimizing over all possible functions. Our aim is to choose an $\mathcal{F}$ (possibly specific to each set of queries) that contains most queries of interest, while still allowing for easy GAN training.

4.2.2 Moment Matching in an RKHS

What functions satisfy the MMD IPM class $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leqslant 1\}$?

For some Reproducing Kernel Hilbert Space $\mathcal{H}$ for a kernel $k(\cdot,\cdot)$, the norm of a function $f \in \mathcal{H}$ is defined as

$$\begin{aligned}
\|f\|_{\mathcal{H}}^2 &:= \langle f, f \rangle_{\mathcal{H}} \\
&= \left\langle \sum_{i=1}^n \alpha_i \phi(x_i), \sum_{i=1}^n \alpha_i \phi(x_i) \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^n \sum_{j=1}^n \alpha_i^2 k(x_i, x_j) \\
&= \alpha^T K \alpha
\end{aligned}$$

where the last equality follows from the fact that all kernel matrices are positive semidefinite. The next question to ask is what is an appropriate choice of RKHS $\mathcal{H}$? More depth

[1] Note that we are able to drop the absolute value sign because by assumption q is closed under negation

Consider the all-subsets kernel over a vector $x \in \{0,1\}^n$, which has a feature ϕ_A for each product of a subset $A \subseteq \{1, 2.., n\}$.

Definition 4.4 (All subsets Kernel([SC]))**.** *The all-subsets kernel ϕ_A is defined by the embedding*

$$\phi : x \rightarrow (\phi_A(x))_{A \subseteq \{1,2..,n\}} \tag{30}$$

with the corresponding kernel $k_{\subseteq}(x, z)$ given by

$$k_{\subseteq}(x, z) = \langle \phi(x), \phi(z) \rangle \tag{31}$$

$k_{\subseteq}$ can be efficiently computed by

$$k_{\subseteq}(x, z) = \prod_{i=1}^{n} (1 + x_i z_i) \tag{32}$$

Thus, our function class $\mathcal{F}$ over the unit ball is

$$\mathcal{F} = \left\{ \alpha \in \mathcal{R}^{|\mathcal{X}|} : \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \left(\prod_{k=1}^{n} (1 + x_{ik} x_{jk} \right) \leqslant 1 \right\} \tag{33}$$

5 QueryGAN

Theorem 3 requires the existence of an actual oracle $\mathcal{O}$ that can find the minimum of a perturbed sum of non-convex functions. In this case, we need an oracle that can minimize the sub of deep neural networks. While SGD is remarkably effective in practice, we are unable to provide guarantees (probabilistic or otherwise) about how close the convergent solution SGD outputs is to the global minima. Recent work suggests that this is not a problem in practice, as spurious local minima (local minima significantly worse than the global minima) get exponentially rarer as the network gets larger [CHM+14]. However, these results still rely on too many impractical assumptions to make them relevant in practice.

summarize approach

Show that GAN objective is concave wrt discriminator paramters ala [GLL+17]

However, while we cannot guarantee (or even certify) convergence to an approximate global minima in general, we can take advantage of the specific structure of the query release problem.

Theorem 5.1. *For any discriminator, the optimal generator G^* has payoff $-V(g, d) = 0$. One such generator draws a uniform sample from the true dataset $\hat{x}$.*

Proof. Recall that the generators payoff is $-V(g, d)$, where the value V is defined as $V(g, d) := \mathbb{E}_{x \sim g, q \sim d} |q(x) - q(\hat{x})|$, where $\hat{x}$ is the true, sensitive dataset. Clearly, the generator's payoff is at most 0. This is trivially attainable if g is the uniform distribution over the rows of $\hat{x}$. ∎

Make clearer the distinction between queries over single rows and queries over all rows,

Maybe another line proving this

Theorem 4 allows us to track how close to optimal the generator G_i is at each step. Assuming D_i is optimal, the regret at each step is simply the generator loss $V(G_i, D_i)$. The cumulative regret is simply the sum of the generator loss at each step.

This gives us a way to track *generator* optimality, assuming the discriminator is optimal. Unlike the generator, there is no obvious maximum payoff for the discriminator in general. However, if we restrict the class of discriminators to single layer neural networks parametrized by θ:

$$D_\theta(x) = \sigma(\theta^T x + b) \tag{34}$$

then $V(G, D_\theta)$ becomes convex with respect to θ, and we can use standard gradient descent methods to guarantee convergence to a global optimum.

Don't reuse σ for thresh-old fn and perturbation

At each round, we update D and G accoring to gradient descent over FTPL. For convenience, let the notation $f_{0:t}$ denote the set of all functions $(f_0, ..., f_t)$.

Prove gradient descent bounds

$$\mathcal{O}_d(\theta_t, f_{0:t}) := \theta_t - \nabla_{\theta_t} \sum_{t=1}^{T} f_t + \sigma_1^T x \tag{35}$$

$$\mathcal{O}_g(\phi_t, g_{0:t}) := \phi_t - \nabla_{\theta_t} \sum_{t=1}^{T} g_t + \sigma_2^T x \tag{36}$$

Algorithm 2: QueryGAN

Input: one-layer discriminator D_θ, deep generator G_ϕ, discriminator and generator oracle $\mathcal{O}_d$, $\mathcal{O}_g$, Rounds T, noise η, output dimension d, game objective V, output rows N

Result: Dataset $x \in \mathcal{X}^d$, Accuracy α

for $t \in 1...T$ **do**

  Draw discriminator and generator perturbations
  $\sigma_1 \sim Exp(\eta)^d$ *and* $\sigma_2 \sim Exp(\eta)^d$

  Update D and G with their respective oracles:
  $\theta_{t+1} \leftarrow \mathcal{O}_d(\theta_t, f_{0:t})$ and $\phi_{t+1} \leftarrow \mathcal{O}_g(\phi_t, g_{0:t})$

  Update losses:
  $f_{t+1}(\cdot) = V(\cdot, D_{\phi_{t+1}})$ *and* $g_{t+1}(\cdot) = V(G_{\theta_{t+1}}, \cdot)$

Calculate cumulative regret: $R \leftarrow \sum_{t \in T} f_t(G_{\phi_t})$

for $i \in 1...N$ **do**

  Draw $t \sim Unif([T])$ and $z \sim \mathcal{N}(0,1)$
  Set $x_i \leftarrow G_{\theta_t}(z)$

return *Dataset* $\{x_1, ..., x_N\}$, *Regret:* R

Theorem 5.2. *Let x, α be the results of running QueryGAN with inputs . Then x is $\alpha-$approximate with respect to all queries representable by D.* What inputs

Proof. □ prove

Explain context, tracking performance

5.1 QueryGAN privacy

Privacy is ensured by the addition of exponentially distributed noise σ_1, σ_2. Interestingly, the original purpose of this noise is not privacy, but rather to ensure convergence of the online algorithm. Because of the deep connections between differential privacy and online learning [NRVW19] [GHM19], however, this noise also ensures ϵ-differential privacy.

5.1.1 Exponential noise

The standard mechanism for ensuring differential privacy is the Laplace mechanism:

Definition 5.1 (Laplace Mechanism [DR13])**.** *Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the laplace mechanism is defined as*

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \ldots, Y_k) \tag{37}$$

where each Y_i are i.i.d drawn from $Lap(\nabla f/\epsilon)$

It's easy to show that the Laplace mechanism preserves $(\epsilon, 0)$-differential privacy [DR13]. However, in *QueryGAN* we add noise drawn from the exponential distribution:

Definition 5.2 (Exponential Distribution). *The exponential distribution with parameter λ is the distribution with density function*

$$Exp(x; \lambda) := \begin{cases} \lambda e^{-\lambda x} & x \geqslant 0 \\ 0 & x < 0 \end{cases} \tag{38}$$

Note that the Laplace mechanism can be seen as the symmetric form of the exponential distribution. Specifically, if $X \sim Lap(\lambda)$, then $|X| \sim Exp(1/\lambda)$.

$$\frac{p_x(x)}{p_x(z)} = \prod \tag{39}$$

use [NRVW19] to show privacy

5.1.2 Tracking privacy loss with moments accountant

Also watch out – if we rely on a fixed T, what if privacy budget is exceeded before we reach it?

5.1.3 Reporting Regret Bounds

- GAN privacy
- Talk about privately reporting α with report noisy max
- Maybe PATE-GAN

5.2 Query Classes

5.2.1 Marginals

While constraining D to be a one layer linear discriminator is restricting, it still is capable of representing a number of query families of interest. Specifically, D contains all k-way marginals.

Definition 5.3 (Marginal). *A marginal $m : \mathcal{X} \to \{0, 1\}$ over a row $x \in \{0, 1\}^m$ is a monotone conjunction, parametrized by some subset S of the input features.*

$$m_S(x) = \prod_{i \in S} x_i \tag{40}$$

We extend this to a dataset X of n rows by defining $m(X) = \sum_{x \in X} m(x)$. A k-way marginal is a marginal restricted to k features.

[DR13]

A k-way marginal can be thought of as counting the number of rows with the same value in the chosen k features. Marginals are a useful way of providing a synopsis of a dataset that still captures complex relationships between features. Producing a differentially private synthetic dataset that agrees with all $k-$way marginals of the true dataset is an extremely well studied problem in the field . .

Survey marginal results

Impossibility results

However, we can show that if $QueryGAN$ succeeds, it is able to match all $k - way$ marginals. This follows from the fact that a linear discriminator can contain all marginals.

This theorem relies on the use of a Rectified Linear Unit activation function

Definition 5.4 (ReLU). *The ReLU activation function $R(x) : \mathbb{R} \to \mathbb{R}^+ := max(0, x)$*

This non-linearity allows us to approximate the nonlinear marginal query with a linear neural network:

Theorem 5.3. *Let D be single layer discriminator parametrized by θ with a ReLU activation function s.t. $D_\theta(x) = \sigma(\theta^T x + b)$. For any single-row marginal m, there exists θ, b s.t. $D_\theta(x) = m(x)$ for all x.*

Show it can be trained to this

Proof. This follows from the definition of a marginal. Let m_S be the marginal over the features S. Let $\theta_i = \mathbb{1}_{i \in S}$. Setting $b = 1 - |S|$, it's clear that

$$D_\theta(x) + 1 = \left\{ \begin{matrix} 1, & \prod_{i \in S} x_i = 1 \\ 0, & \text{otherwise} \end{matrix} \right\} = m_S(x) \tag{41}$$

□

Theorem 5.4. *Let G, α be the output of running QueryGAN with . Then G will generate a dataset with all marginal counts accurate to within α.*

Parameters, steps etc

The proof of this theorem follows directly from Theorem 5.2 and Theorem 5.3.

Oracle runtime (also comment on non-oracle runtime)

Summary, contextualize empirical results

6 Practical QueryGAN Heuristics

6.1 Stochastic Loss Subsampling

Even if we assume that the offline optimization oracle $\mathcal{O}$ is in general good at optimizing non-convex functions, note that at each step t *QueryGAN* requires $\mathcal{O}$ to optimize over *sums* of multiple highly non-convex neural networks

$$\arg\min_x \sum_{t=1}^{T} g_t(x) + \sigma_2^T x \tag{42}$$

Thus, as T grows larger in later rounds, this optimization problem becomes dramatically larger and less feasible in practice, straining our oracle assumptions.

One might hope that we can obtain almost as good regret guarantees by instead running our oracle over a random sample of loss functions drawn from some distribution over past loss functions.

Algorithm 3: Subsampled Follow the Perturbed Leader (S-FTPL)

Input: Noise $\eta > 0$, Rounds T, Loss Distribution $L(f_{1:t})$
Result: Actions $x_{1:T}$
for $t \in 1 \ldots T$ **do**
  Draw i.i.d random vector $\sigma \sim Exp(\eta)^d$
  Sample batch of loss functions $B \sim L(f_{1:t})$
  Prediction at time t:

$$x_t \leftarrow \arg\min_{x \in \mathcal{X}} \sum_{f \in B} f(x) - \sigma^T x$$

Theorem 6.1. *If S-FTPL is instantiated with loss distribution ???, then the strategies $x_{1:t}$ ensure sublinear regret*

Proof. □

6.2 QueryGAN Rejection Sampling

While we can prove sufficient regret bounds simply by sampling uniformly from $G_{1:t}$, this naive methods throws away valuable information. Specifically, the discriminators $D_{1:t}$ are able to evaluate the quality of a generated sample. If we train

citations

6.3 Tailored Loss Functions

7 QueryGAN with alternate discriminators

While QueryGAN follows the standard GAN practice of representing both the generator and discriminator with a neural network, this is not mandatory. Indeed, given that D is represented by the almost trivially simple 1 layer network, D is best understood in more general terms than a neural network.

Definition 7.1 (Tractable Discriminator Set). *We say a class of functions $\mathcal{F} : \mathcal{X} \to \{0, 1\}$ parametrizes a set of tractable discriminators w.r.t a class of queries $\mathcal{Q}$ iff*

1. $\mathcal{Q} \subseteq \mathcal{F}$
2. *There exists a tractable offline oracle $\mathcal{O}$*

What does it do (also define tractable)

As shown above, the set $\mathcal{F}_{single}$ of one layer neural networks is a tractable discriminator set for all marginals (as well as all sigmoided linear functions in general). This has the benefit of being easy to optimize in practice, without the runtime depending exponentially on the dimensionality of the query space. Relaxing that restriction by allowing for less efficiently optimizable discriminators lets us generate α-accurate synthetic data for much larger query classes.

7.0.1 Multiplicative Weights

Consider the application of the renowned Multiplicative Weights algorithm to the query release problem, introduced in [HR10].

Definition 7.2 (Multiplicative Weight Oracle Algorithm). *Fix a class of queries $\mathcal{Q}$ and a true dataset $\hat{x}$. Define an initial uniform distribution over queries θ_0. Let $\mathcal{O}_{MW}(\theta_t, G_t)$ output a reweighted distribution*

$$\theta_{t+1}^q \propto \exp\left(-\eta V(G_t, q)\right) \cdot \theta_t^q \tag{43}$$

for each $q \in \mathcal{Q}$

θ_t defines a distribution, with each query weighted proportionally to how well it distinguishes real data from fake data. The discriminator draws $D_{\theta_t}(x) := |q(x) - q(\hat{x})$ where q is a query drawn $q \sim \theta_t$.

7.1 TODO

- Local DP GAN
- FTPL oracle comes with privacy for free – but also talk about how to handle non-private oracle

Proofs:

- GAN objective =? query release objective
- FTPL gradient descent (not clear this is necessarily a full on proof)
- Proof of accuracy of subsampled FTPL objective

7.2 Adversarial Kernel Learning

Limiting

8 Empirical Results

9 Conclusion

References

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. January 2017.

[AGH18] Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in Non-convex Games with an Optimization Oracle. *arXiv:1810.07362 [cs, stat]*, October 2018.

[CHM+14] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. *arXiv:1412.0233 [cs]*, November 2014.

[DR13] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[GAH+14] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual Query: Practical Private Query Release for High Dimensional Data. *arXiv:1402.1526 [cs]*, February 2014.

[GHM19] Alon Gonen, Elad Hazan, and Shay Moran. Private Learning Implies Online Learning: An Efficient Reduction. *arXiv:1905.11311 [cs, stat]*, May 2019.

[GLL+17] Paulina Grnarova, Kfir Y. Levy, Aurelien Lucchi, Thomas Hofmann, and Andreas Krause. An Online Learning Approach to Generative Adversarial Networks. *arXiv:1706.03269 [cs, stat]*, June 2017.

[GPM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[GXC+18] Hao Ge, Yin Xia, Xu Chen, Randall Berry, and Ying Wu. Fictitious GAN: Training GANs with Historical Models. *arXiv:1803.08647 [cs, stat]*, March 2018.

[Haz19] Elad Hazan. Introduction to Online Convex Optimization. *arXiv:1909.05207 [cs, math, stat]*, September 2019.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A Simple and Practical Algorithm for Differentially Private Data Release. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2339–2347. Curran Associates, Inc., 2012.

[HR10] M. Hardt and G. N. Rothblum. A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, October 2010.

[HRU13] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential Privacy for the Analyst via Private Equilibrium Computation. *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing - STOC '13*, page 341, 2013.

[JY19] James Jordon and Jinsung Yoon. PATE-GAN: GENERATING SYNTHETIC DATA WITH DIFFERENTIAL PRIVACY GUARANTEES. page 21, 2019.

[LSZ15] Yujia Li, Kevin Swersky, and Richard Zemel. Generative Moment Matching Networks. *arXiv:1502.02761 [cs, stat]*, February 2015.

[Mül97] Alfred Müller. Integral Probability Metrics and Their Generating Classes of Functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

[NRVW19] Seth Neel, Aaron Roth, Giuseppe Vietri, and Zhiwei Steven Wu. Differentially Private Objective Perturbation: Beyond Smoothness and Convexity. *arXiv:1909.01783 [cs, stat]*, September 2019.

[NRW18] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to Use Heuristics for Differential Privacy. *arXiv:1811.07765 [cs, stat]*, November 2018.

[SC] John Shawe-Taylor and Nello Cristianini. Kernel Methods for Pattern Analysis. page 478.

[SN19] Arun Sai Suggala and Praneeth Netrapalli. Online Non-Convex Learning: Following the Perturbed Leader is Optimal. *arXiv:1903.08110 [cs, math, stat]*, March 2019.

[Vil08] Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

[ZML17] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based Generative Adversarial Network. *arXiv:1609.03126 [cs, stat]*, March 2017.

[zot] Two-Sample Tests, Integral Probability Metrics, and GAN Objectives. http://www.gatsby.ucl.ac.uk/~dougals/slides/dali/#/63.