The Dissertation Committee for Maurice Diesendruck
certifies that this is the approved version of the following dissertation:

Distribution Distance Measures in Generative and Privacy Models

Committee:

Sinead A. Williamson, Supervisor

Mingyuan Zhou, Supervisor

Stephen Walker

Lizhen Lin

Distribution Distance Measures in Generative and Privacy Models

by

Maurice Diesendruck, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2019

Dedicated to my family.

Acknowledgments

I wish to thank the multitudes of people who helped me.

Distribution Distance Measures in Generative and Privacy Models

Publication No. _____

Maurice Diesendruck, Ph.D.
The University of Texas at Austin, 2019

Supervisors: Sinead A. Williamson
Mingyuan Zhou

Distribution distance measures provide a useful class of tools for generative and privacy models. In both cases, the goal is to simulate a data distribution without revealing too much about individual points. While early generative models focused on matching data in a component-wise manner, the models in this work incorporate distribution metrics to provide population-level information during training. Doing so reduces overfitting and increases the model's ability to generalize. Maximum mean discrepancy and energy distance are two such metrics that are easily defined and implemented over samples, and provide meaningful results on a range of data sets and data types. This work presents three main contributions: (1) a novel use of importance weights to modify the output distribution of a generative model, (2) an application and evaluation of a generative model for medical data privacy, and (3) a novel method for private data synthesis using support points and differential privacy.

Table of Contents

Acknowledgments **v**

Abstract **vi**

List of Tables **xi**

List of Figures **xii**

Chapter 1. Introduction **1**

Chapter 2. Background **4**

2.1 Distribution Distance Measures 4

2.1.1 f-Divergences 4

2.1.2 Integral Probability Metrics 5

2.1.3 Maximum Mean Discrepancy 5

2.1.4 Support points 7

2.2 Neural Networks and GANs 7

2.2.1 Deep Neural Networks 7

2.2.2 Generative Adversarial Modeling 8

2.3 Differential Privacy 9

2.3.1 Standard Model 10

2.3.2 Interactive and Sequential Queries 12

2.3.3 Noise mechanisms 13

2.3.4 Sensitivity calculations 15

2.4 Differentially private data release 17

Chapter 3. Importance Weighted Generative Networks **19**
3.1 Introduction 19
3.1.1 Related Work 21
3.2 Problem Formulation and Technical Approach 24
3.2.1 Maximum Mean Discrepancy between Two Distributions 25
3.2.2 Importance Weighted Estimator for Known M 25
3.2.3 Robust Importance Weighted Estimator for Known M . 27
3.2.4 Self-normalized Importance Weights for Unknown M . . 28
3.2.5 Approximate Importance Weighting by Data Duplication 29
3.3 Evaluation 29
3.3.1 Can GANs with Importance Weighted Estimators Recover Target Distributions, Given M? 31
3.3.2 In a High-dimensional Image Setting, How Does Importance Weighting Compare with Conditional Generation? 33
3.3.3 When M Is Unknown, but Can Be Estimated Up to a Normalizing Constant on a Subset of Data, Are We Able to Sample from our Target Distribution? 34
3.4 Conclusions and Future Work 36

Chapter 4. Synthetic Medical Records **37**
4.1 Problem Statement 37
4.2 GAN Pre-Processing 37
4.3 Implementation Details 38

Chapter 5. Support Points and Privacy **41**
5.1 Introduction 41
5.2 Background 42
5.2.1 Differential Privacy 42
5.2.2 Energy Distance 44
5.2.3 Support Points 45
5.3 Sampling Private Synthetic Sets via the Exponential Mechanism with Energy Distance 45
5.3.1 Motivation and Summary 45
5.3.2 Sensitivity 46

5.3.3 Worked Example 48
5.3.4 Sampling Synthetic Sets 49
5.4 Practical Private Synthesis using Support Points 50
5.4.1 Support Point Method and Sensitivity 51
5.4.2 Algorithm 53
5.5 Convergence 53
5.6 Expansion of Support Points to Full Synthetic Data Set 56
5.6.1 Kernel Density Estimation 56
5.6.2 Repeated Sampling 57
5.7 Experimental Results 57
5.7.1 Gaussian Mixture Model 57
5.7.1.1 Data 58
5.7.1.2 Support Points Optimization 58
5.7.1.3 Sample using Metropolis-Hastings 58
5.7.1.4 KDE with Pre-Selected Bandwidth 60
5.7.1.5 KDE with DP-MLE Bandwidth 60
5.7.1.6 Repeated Sampling 62
5.7.2 Regression Performance 63
5.8 Related work 65
5.9 Conclusion 66

Chapter 6. Conclusion **69**

Appendices **70**

Appendix A. Importance Weighted Generative Networks **71**
A.1 Proof of Theorem 1 71
A.2 Proof of Theorem 2 72
A.3 Implementation and Additional Experiments 74
A.3.1 Synthetic Data 74
A.3.2 Yearbook 74
A.3.3 MNIST 74

Bibliography **82**

Index **98**

Vita **99**

List of Tables

3.1 Constructing importance weighted estimators for losses involving U-statistics, V-statistics and sample averages. Here, $\mathcal{U}$ is the set of all r-tuples of numbers from 1 to n without repeats, and $\mathcal{V}$ is the set of r-tuples allowing repeats. Below, let $X_{u,*} = X_{u_1}, ..., X_{u_r}$. 30

3.2 Distributional discrepancies between generated and target data samples 32

A.1 Distributional discrepancies between generated and target data samples 75

A.2 Minimum distributional discrepancies between generated and target data samples 76

List of Figures

2.1 For a counting query where the true response is 108, and any single-row change could maximally change the response to 107, the orange distribution represents the noise distribution from which to sample a differentially private result. The privacy level is defined by the maximum density ratio between the orange and blue distributions. 13

2.2 Data x is partitioned into m subsets. Query f is performed on each subset producing outputs $z_1, \ldots, z_m$. Outputs are aggregated using function g, and differentially private noise is added based on the sensitivity of the aggregation. 17

3.1 If our target distribution $\mathbb{P}$ differs from our observed distribution $M\mathbb{P}$, using the standard estimator will replicate $M\mathbb{P}$, while an importance weighted estimator can replicate the target $\mathbb{P}$. 20

3.2 Importance weights are used to accurately rebalance an uneven class distribution. 32

3.3 Example generated images for all example networks, Yearbook dataset [48]. Target distribution is uniform across half-decades, while the training set is unbalanced. 33

3.4 Partial labeling and an importance weighted estimator boost the presence of sevens with horizontal bars. In 3.4a and 3.4b, samples are sorted by predicted weight, and in 3.4c, the empirical CDFs of data, generated, and importance duplicated draws, are shown, where the latter serves as a theoretical target. The generated distribution is close in distance to the target. 35

4.1 GAN-preprocessed EHR data versus raw EHR data. (a) Marginal distribution of each variable. For each variable, the two overlaid histograms show the agreement between the preprocessed and the raw data. The variable names and ranges are deliberately not shown. (b) Correlation of each pair of variables. 39

5.1 Support points (green) accurately capture complex clustering structures in data (gray) over a range of cluster variance and cluster count. In each setting, 200 data points are represented by 10 support points. 59

5.2 As the privacy budget α increases, privately sampled support points (red) more accurately represent data (gray) by moving closer to true support points (green). In each setting, 200 data points are represented by 10 private support points. 59

5.3 As a baseline, the Metropolis-Hastings procedure is tested under a minimally private setting with $\alpha = 5000$. The sampling scheme mixes well, and captures the distribution. 60

5.4 Full samples (blue) from a kernel density estimator centered on privately sampled support points (red), using various preselected bandwidths. The result approximates the data distribution (gray). Results are shown for $4\sigma, \sigma$, and $\sigma/4$. 61

5.5 A private support point set can be expanded using a kernel density estimator, using a private version of the maximum likelihood bandwidth. 62

5.6 Repeated samples of support points can be collected to yield a full sized synthetic data set. Since the privacy budget accumulates with each repetition, repeated samples might be useful for settings with low sensitivity, where individual samples maintain utility. 63

5.7 Metropolis-Hastings convergence diagnostics show multiple, randomly initialized chains converging around a mean energy distance. The Gelman-Rubin statistic trends toward one, and between-chain variance trends toward zero. The example illustrated is for support point size 40 and privacy budget 1000. 65

5.8 Private support points outperform similar methods in all but the lowest-budget setting (where performance is on par), and comparative advantage grows with dimensionality. Error bars and central dots represent interquartile ranges and medians, respectively. *A random subset of 500 points is chosen for training in the California dataset. 68

A.1 Example interpolations in the latent z space, half-decades experiment. 77

A.2 Example generated yearbook images from two time periods: Old (1930) and Recent (1980-2013). The target distribution is 50%/50%, while the training set is 1%/99%. Again, C-DCGAN is unstable across a variety of training parameters, while the importance weighted MMD-GAN methods produce reasonable samples (b)–(d) with meaningful interpolations in the latent space (f)–(h). 78

A.3 Example generated yearbook images from two time periods: Old (1925-1944) and Recent (2000-2013). Target distribution is 83%/17% while the given data $M\mathbb{P}$ is split 50%/50%. Each time period contains enough images to train C-CDGAN successfully. However, the other methods produce qualitatively sharper images (a)–(d) with smoother latent interpolations (e)–(h). . . 80

A.4 Importance weights are used to accurately boost an even class distribution to specified levels. 81

A.5 A small set of labels are used to train an importance weighted estimator that aims to boost the presence of flat-bottomed twos. In A.5a and A.5b, samples are sorted by predicted weight, and in A.5c, the empirical CDFs of data, generated, and importance duplicated draws, are shown, where the latter serves as a theoretical target. The generated distribution produces more flat-bottomed twos, and is close in distance to the target, with $d_{KS} = 0.07$, $p = 0.376$. 81

Chapter 1

Introduction

How should information about a data set be communicated? Not surprisingly, many options exist. Three categories of response might apply. A scientist could share: (1) Summary statistics, (2) a probability distribution function with specific parameter values, or (3) a set of representative points or *synthetic data*. Summary statistics provide easy-to-compute descriptions of data, but can be too reductive. Probability distribution functions provide a model to generalize behavior and even to sample new points, but can be intractable to compute for complex phenomena. Synthetic data, while not perfect, provide an intriguing combination of descriptiveness and flexibility in representing complex distributions. This work explores the generation and utility of synthetic data, and the challenges presented in their application.

Synthetic data can be valuable in many settings. Consider practical applications where more data is needed, for example, to populate characters in a video game or rows in a testing database. Alternatively, suppose the data is biased in a known way, and a corrected sample could be generated. Finally and perhaps most commonly, consider the case in which a data owner wants to share data, but not disclose individual information. In each setting, it can be useful to produce similar, but not identical, new points that can act as a direct replacement for original data.

Many methods aim to accomplish this task. Generative adversarial networks [50] (GANs) use neural networks to learn a mapping between noise and data distributions, and can produce an arbitrary number of new points. GANs have been used in complex settings, creating synthetic databases and even synthetic images and audio. Bootstrapping expands a data set by resampling with replacement, and is extremely simple to implement. Density estimation includes a wide class of methods to identify a probability distribution function from which samples can be drawn. Among these methods,

several make use of distribution metrics to provide population-level, as opposed to individual-level information. This work continues in the spirit of distribution-matching, using distribution metrics to address certain challenges in applications of synthetic data.

Chapter 3 of this work addresses the challenge of generating synthetic data when the data distribution contains a known bias, or needs to be modified. Two related fields are domain adaptation and conditional models. In domain adaptation, weights are applied to classification or regression methods to accommodate a test set that is distributed differently from a training set. In conditional models, a practitioner learns conditional distributions and subsequently samples using a desired distribution of conditional values.

This chapter presents methods to accommodate modifications between data and sampling distributions via *importance weighting*, and allows one to estimate a loss function with respect to a target distribution even if that target is not directly accessible. These estimators, which differentially weight the contribution of data to the loss function, offer theoretical guarantees that heuristic approaches lack, while giving impressive empirical performance in a variety of settings. The methods presented accommodate arbitrary target distributions, adding flexibility compared to domain adaptation methods, and accommodate modifications over continuous-valued latent features, adding flexibility compared to conditional methods.

Chapter 4 of this work addresses the challenge of disclosure of individual-level information. Disclosure can occur in a number of settings, and for a variety of models. For example, [125] show that medical information can be linked to individuals using only census-level information (e.g zip code and birth date) in public databases. Later, [118] showed that an individual's presence in a training set can be extracted by comparing predictions between models trained with and without that point — a class of methods called "membership inference attacks".

This chapter synthesizes medical data using a GAN model that specifically minimizes distributional distance, then evaluates privacy performance based on two known privacy risks: presence and attribute disclosure. The

results demonstrate that a close distributional match is possible without replicating data exactly, and that presence disclosure can be low. Results for attribute disclosure represent higher risk levels, but not significantly above a baseline level.

While the results in Chapter 4 are evidence that GAN models can sometimes produce synthetic data that satisfy some privacy risks, these models do not offer guarantees. Chapter 5 of this work addresses the challenge of generating synthetic data in way that offers formal, measurable, and robust privacy guarantees. *Differential privacy* is a specific privacy procedure often used because of its prescriptive nature and its protection against arbitrary side information (as different from the medical data leak that occurred by linking public, census-style databases). Differential privacy adds random noise to data, or to the response of a query over data, so that variations in response will mask the presence or absence of any single point. Formally, it provides a bound on the likelihood ratio of any response computed with or without any single point.

This chapter demonstrates a novel method of differentially private data synthesis using a representative point set called support points, and a noise mechanism called the exponential mechanism. A theoretically valid and easily implemented Metropolis Hastings sampling procedure demonstrates the anticipated behavior, where larger and less-private synthetic sets perform well as substitutes for data, and smaller and more-private sets perform worse.

Chapter 2

Background

2.1 Distribution Distance Measures

Distribution distance measures quantify the dissimilarity between empirical data sets or between probability distributions. Two general mathematical structures exist to derive such quantities, called f-divergences or ϕ-divergences, and integral probability metrics. While all integral probability metrics satisfy the four criteria of metrics – non-negativity, symmetry, identity, and triangle inequality – some divergences do not. The Kullback-Leibler divergence, for example, does not satisfy the symmetry property of metrics, though the square root of a related measure called the Jensen-Shannon distance does.

2.1.1 f-Divergences

The family of f-divergences measures the weighted sum, taken over the shared support of two distributions, of a convex function of the ratio of two probabilities. [120] provides background, and use the following notation:

$$D_\phi(\mathbb{P}, \mathbb{Q}) := \int_M \phi\Big(\frac{d\mathbb{P}}{d\mathbb{Q}}\Big) d\mathbb{Q}, \quad \text{for } \mathbb{P} \ll \mathbb{Q}, \text{ otherwise } +\infty, \tag{2.1}$$

for measurable space M where ϕ is convex and $\phi(1) = 0$. Several common measures fall into this category, such as the Kullback-Leibler divergence with $\phi(t) = t \log t$, total variation distance with $\phi(t) = |t-1|$, and Hellinger distance with $\phi(t) = (\sqrt{t}-1)^2$. These measures are akin to a weighted sum of likelihood ratios between $\mathbb{P}$ and $\mathbb{Q}$.

2.1.2 Integral Probability Metrics

The family of integral probability metrics (IPMs) measure the largest possible difference between expected values of functions over each distribution. In the notation of [120], IPMs are defined as:

$$\gamma_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \left| \int_M f d\mathbb{P} - \int_M f d\mathbb{Q} \right| \tag{2.2}$$

for family $\mathcal{F}$ of bounded measurable functions. Since these functions can accentuate certain qualities of each probability distribution, the IPM therefore serves as a "worst-case difference", as observable through that family of functions. When the family is 1-Lipschitz, $\{\mathcal{F} = f : ||f||_{\mathcal{L}} \leq 1\}$, this corresponds to the Kantorovich metric, and when the measure space M is also separable, this corresponds to the well-known Wasserstein distance.

2.1.3 Maximum Mean Discrepancy

When the function family $\mathcal{F}$ is restricted to the unit ball of a reproducing kernel Hilbert space (RKHS), $\{\mathcal{F} = f : ||f||_{\mathcal{H}} \leq 1\}$, this corresponds to the maximum mean discrepancy (MMD). For samples X and Y of size N and n, respectively, the MMD is defined as

$$\text{MMD}^2(X, Y) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} k(x_i, x_j) - \frac{2}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} k(x_i, y_j) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(y_i, y_j), \tag{2.3}$$

where the collection of pairwise kernel values is sometimes called the Gram matrix. The MMD was originally proposed in [52] and used in a hypothesis test to answer the two-sample problem of whether samples came from the same distribution. Since the metric can be evaluated over samples, it is sometimes called "distribution-free", unlike the divergence measures that require functional forms of probability distributions. The following related concepts enable the theoretical utility of the MMD.

U-statistics and V-statistics When estimating the MMD with empirical samples, the manner of calculation affects whether the estimator is biased.

For a measure such as MMD which takes expectations over Gram matrices, the asymptotically unbiased U-statistic excludes diagonal terms, while the V-statistic includes them. Foundational asymptotic results are due to [59], which provides definitions for estimators based on samples.

Reproducing kernel Hilbert space The RKHS is defined by a positive definite kernel [99], and provides an embedding based on an infinite set of basis functions. If the kernel is *characteristic*, then the RKHS embeddings of two distributions will be equal if and only if the distributions are identical.

Characteristic kernel As originally described in [44], the kernel is characteristic if the mean map from sample to Hilbert-space mean element is injective.

Kernel bandwidth selection The choice of kernel bandwidth is critical to performance for MMD methods. In [53], for example, the kernel choice is selected to optimize power in a two-sample hypothesis test.

Fast approximations to MMD Due to the pairwise computations inherent to the Gram matrix of the MMD, various faster approximations have been explored. In [86], experiments demonstrate the utility of a low-dimensional randomized features approach, originally proposed by [111]. Building on this approach, [144] present a related method using results from harmonic analysis and shift-invariant kernels.

Energy distance The energy distance is a special case of the MMD, with kernel equal to the negative Euclidean norm between elements. Defined in [127], the energy distance presents a simple measure that can be flexibly applied given a measure space on elements of high dimension.

2.1.4 Support points

In resource-constrained environments or when given highly correlated data, it can be useful to reference only summaries of large data sets. This classical problem has spurred a handful of data reduction methods including coresets, support points, and clustering.

Support points [91] are a small and representative set of points which minimize the energy distance to a larger data set. The energy distance can be evaluated over finite sample sets, making it appealing for practical applications where distributions do not have a known functional form. Consider data $X = \{x_1, \ldots, x_N\}$ on space $\mathcal{D} \in [0,1]^d$. The optimal set of support points $Y = \{y_1, \ldots, y_n\}$ minimizes the following expression in term of Y:

$$e_{N,n}(X,Y) = \frac{2}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n}||x_i - y_j||_p - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}||x_i - x_j||_p - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}||y_i - y_j||_p, \tag{2.4}$$

with $p = 2$. Generalized distance metrics in place of the $||\cdot||_2$ Euclidean norm also hold under certain conditions [88].

2.2 Neural Networks and GANs

In the past four decades, neural networks have been developed from first principles into powerful, general function-learning tools with applications in complex human perception. [16] describes the historical beginnings from even earlier notions of an artificial neuron as a neurophysiologically inspired boolean logic gate [94], to the first single-layer neural network in the form of Restricted Boltzmann Machines [119] and autoassociators [116].

2.2.1 Deep Neural Networks

In the mid-1980s, researchers interested in artificial intelligence began to assemble layers of artificial neurons to build more complex functions. Deep Belief Networks and stacked autoassociators were proposed and became the basis of the "hidden layers" of deep neural networks. The word "deep" describes how in certain complex settings, a large number of layers were found

to be effective. In such models, each layer was intended to be a representation of its input, possibly in an abstract sense. By applying an affine transformation followed by a non-linear function, unusual logic could be encoded. In the image setting, the notion of a convolutional neural network [69] was proposed. Instead of making decisions based on raw pixel values, pixel values would be transformed using filters that act as feature detectors and sweep across the image input. Each filter is a randomly initialized matrix, which when multiplied with a section of the image, yields a signal for the presence of that feature. The image is therefore transformed into a numerical representation based on filter values. The innovation of [69] was to structure the network of filters and demonstrate the optimization method called "backpropagation", which enabled meaningful results in complex domains such as computer vision and speech recognition.

The idea that components of the model should capture different features led to the language of "feature maps" [69] and "distributed representation" [16]. The deep neural network is designed to be a hierarchy of components, where each layer or filter captures a different feature, or different level of the overall understanding of the input. In image networks, features can correspond to low-level textures [46, 47] or even to semantically complex concepts like "outdoors" [128, 130].

2.2.2 Generative Adversarial Modeling

Many statistical modeling questions are expressed in the form of a clustering, classification, or regression. Given an input, the model seeks to assign an associated label or value. A typical model setup involves a set of labeled data, a model which takes input and predicts output, and a loss function which scores the performance of the model's predicted outputs compared to true labels. Consider a meteorological model that predicts the amount of precipitation tomorrow, based on the today's temperature, wind, humidity, and pressure. A regression model will predict an amount, and the loss function would register the amount of error relative to the true value in the data. Optimizing such a model involves adjusting its parameters in a way that brings predictions closer to the truth. In a clustering or classification problem, the model also adjusts parameters so that objects in the same cluster or class are

close together, while objects in different clusters are far apart. In all cases, the model adjusts in order to transform the meaningful features of the input into a signal that performs reliably for the task.

In generative adversarial modeling, information extraction is achieved by means of reproducibility. In a *learn-by-doing* sense, the generative models claims that if it can reproduce similar objects, then it will have captured the meaningful features of the data distribution. For neural networks, this idea was proposed in [50] as Generative Adversarial Networks (GANs), where a generative model makes simulations [of data] and a classifier learns to distinguish simulations from data. When trained together, the generative function and the classifier can dynamically improve each other, until the generative function produces simulations indistinguishable from data.

2.3 Differential Privacy

Practitioners often want to share data publicly, while also respecting the privacy of individuals. In an effort to preserve privacy, data stewards will anonymize, redact, summarize, randomize, or add noise to data before publication, in order to make it hard for an adversary to identify individuals. Privacy breaches can lead to financial, political, and personal losses, and have warranted the attention of large businesses and institutions. In most discussions, researchers accept a trade-off between privacy and utility. Intuitively, the most private summary of data is a uniform distribution over its support – it is also the least informative, because an adversary cannot identify an individual with better than random chance. The private distribution becomes more informative as it shifts away from the uniform distribution and toward the data.

In [133], Wagner and Eckhoff present a survey of privacy metrics in eight broad categories, each related to the effect of a disclosure to the public. Most categories measure a change that occurs after a disclosure, and imply that more change means more information and less privacy. The categories are: 1. adversary uncertainty about an individual, 2. information gain due to disclosure, 3. data similarity between true and disclosed data, 4. indistinguishability between individuals in the disclosure, 5. adversary success rate,

6. prediction error, 7. amount of time until adversary success, and 8. adversary accuracy and precision. While most measures are descriptive, one of the indistinguishability metrics called *differential privacy* is prescriptive. Rather than passively measuring the change due to disclosure, it actively induces an upper-limit on change as a function of noise added to data.

The method of differential privacy is particularly appealing because it allows a data steward to make mathematical guarantees about their disclosure, and protects against arbitrary side-information and linkage attacks [39]. It is usually presented to individuals as a reason to feel comfortable when contributing data to a study or company. From this perspective, privacy is ostensibly available wherever and to whatever degree it is demanded. As a result, a large class of privacy research has flourished, focusing on ways to induce differential privacy guarantees, while preserving as much utility as possible.

2.3.1 Standard Model

The promise of data collection is the ability to learn underlying behavior in a way that generalizes to new data. If an underlying distribution were known, it could be communicated directly without disclosing *any* individual information. Since this is not available in practice, information is derived from finite samples, which are the best-available expression of the true distribution. A private data set is a similar distribution that provides individual privacy protection.

Consider the standard model of differential privacy as defined by Dwork in [36]. A randomized function $\mathcal{M}(D) = f(D)+\eta$, sometimes called a sanitizer [38], induces a distribution around a deterministic, true function response. By definition, $\mathcal{M}$ gives α-differential privacy if for all data sets D and D' differing in at most one element ("neighboring sets"), and all $\nu \subseteq Range(\mathcal{M})$,

$$\frac{\Pr[\mathcal{M}(D) \in \nu]}{\Pr[\mathcal{M}(D') \in \nu]} \leq \exp(\alpha). \tag{2.5}$$

i.e. the density ratio for observing any randomized function output is bounded.

In the simplest case known as the *Laplace mechanism*, $\mathcal{M}$ adds zero-mean Laplace noise to the function response, and the density ratio bound is derived as follows:

$$\frac{f(D') + \text{Lap}(0,\lambda)}{f(D) + \text{Lap}(0,\lambda)} = \frac{\text{Lap}(f(D'),\lambda)}{\text{Lap}(f(D),\lambda)} = \frac{\exp(\frac{-|y-f(D')|}{\lambda})}{\exp(\frac{-|y-f(D)|}{\lambda})} = \exp\Big(\frac{|y-f(D)| - |y-f(D')|}{\lambda}\Big) \tag{2.6}$$

$$\leq \exp\Big(\frac{|f(D)-f(D')|}{\lambda}\Big) \leq \exp\Big(\frac{\Delta f}{\lambda}\Big) := \exp(\alpha), \tag{2.7}$$

where Δf is called the *sensitivity*. The sensitivity represents the largest possible difference in function responses over pairs of neighboring data sets, i.e. $\Delta f = \max_{D,D'} |f(D) - f(D')|$. [1] Though the notation here reflects a scalar query output, privacy conditions hold equivalently for vector-valued outputs, where absolute value would be replaced by the appropriate vector norm.

The nature of this bound, with a likelihood ratio, is reminiscent of the form of divergence measures. The relationship is formalized in Definition 3.6 of [39], where differential privacy is shown to be equivalent to an upper bound on max-divergence. The bound can also be viewed as a Lipschitz condition on the Hamming metric on the space of the data set, where for metric $\mathrm{d}_H(\cdot,\cdot)$, sensitivity $\Delta f = |f(D) - f(D')|/\mathrm{d}_H(D,D')$ [38].

When communicating the level of privacy, α is typically given, and takes several names in the literature: privacy level, privacy budget, leakage, or *privacy loss*. The privacy practitioner aims to add enough noise to attain a specific, ideally low privacy loss, while maximizing the utility of the resulting distribution.

The function f often belongs to a class of functions called "predicate queries" [20], where the query returns the proportion of the data set that satisfies some set of predicates, or conditions. Since the proportion is a count divided by the data size, such queries are similarly called count, range-count, linear counting, histogram, and contingency table queries [78, 39, 11]. In a database context with a query that counts rows satisfying a condition, the term $|f(D) - f(D')|$ is at most one, since changing one row could maximally change the number of valid rows by one.

[1] A similar approach exists with Gaussian noise and an l_2 version of the sensitivity. There is some debate whether there are substantial theoretical advantages [39, 109].

Queries can also involve more complex information, such as the interquartile range [37], order statistics [103], and full databases [20, 73]. In each case, the amount of added noise is tailored to the sensitivity of the information that must be accessed to produce the answer. The following section describes the components of differential privacy theory that are specific to systems allowing repeated queries, and systems allowing unlimited queries on pre-privatized, synthetic data.

2.3.2 Interactive and Sequential Queries

Data release can be interactive and relatively narrow, via repeated summary-style queries over a database; or can be non-interactive and full, via a one-time anonymized disclosure. When repeating queries, the following theorems apply to the total privacy loss.

Composition theorems The sequential composition theorem states that privacy losses over sequential queries accumulate additively. That is, for k queries on data set D that are each α-differentially private, the total sequence of queries is $k\alpha$-differentially private. Such privacy degradation make intuitive sense, since repeated noisy queries will eventually converge to the truth [39, 96]. Tighter bounds on k-fold composition have since been proposed in [67]. The parallel composition theorem states that application of α_k-differentially private queries on each subset D_k in a partition of D, yields $\max(\alpha_k)$-differential privacy [38, 96]. Both theorems can be important when assessing how a user's interaction with private data will affect the overall privacy loss.

Post-processing The notion of immunity to post-processing, described in Proposition 2.1 of [39], is also critical to the design of private systems. It states that applying arbitrary deterministic functions to differentially private results does not affect privacy loss. Intuitively, this demonstrates that once information has been obscured, it is impossible to reveal additional information through extended investigation.

2.3.3 Noise mechanisms

Noise mechanisms induce differential privacy by adding noise to query responses. For neighboring data sets D and D' on data space $\mathcal{X}$, let $f : \mathcal{X} \to \mathbb{R}^k$ be any real-valued query function. Consider that adding noise to query responses is akin to describing two kernel distributions centered at $f(D)$ and $f(D')$, respectively. When $f(D)$ and $f(D')$ are far apart and kernel distributions are narrow, the maximum density ratio can be large. When $f(D)$ and $f(D')$ are close and kernel distributions are wide, the maximum density ratio can be close to one. Figure 2.1 from [96] illustrates this concept. In the design of differential privacy mechanisms, the scale of the noise mechanism is tuned to give the desired privacy level, given the sensitivity of the query.

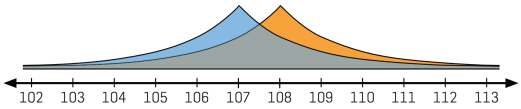


Figure 2.1: For a counting query where the true response is 108, and any single-row change could maximally change the response to 107, the orange distribution represents the noise distribution from which to sample a differentially private result. The privacy level is defined by the maximum density ratio between the orange and blue distributions.

Exponential mechanism For query responses in discrete or categorical space, adding continuous numeric noise might not apply. Consider the question "What is the most common name in the database?" There is no clear sense of how to add noise in this setting. Another classic example is privatizing individual bids in an auction – where the auction's revenue can be related to price in a complex way – and bids must be credible after privatization. At a certain point, a price increase might drastically change the expected revenue, making it an unreasonable value. In these cases, a utility function acts as a link between the output space and a space that varies more smoothly, in which noise can be added. This procedure is called the exponential mechanism [95]. In the case of the auction, the utility function links the price space to the revenue space, and sampling occurs with respect to the revenue space.

Formally, the exponential mechanism defines a utility function on the joint space of query inputs and outputs, and samples outputs with probability exponential in their utility score. For a utility function $\mu : \mathcal{X} \times Range(\mathcal{M}) \to \mathbb{R}$, the sensitivity is defined as

$$\Delta u = \max_{\nu \in Range(\mathcal{M})} \max_{D,D'} |u(D,\nu) - u(D',\nu)|, \tag{2.8}$$

and the α-differentially private sampling distribution is defined as

$$\Pr[D] \propto \exp\Big(\frac{\alpha\, u(D,\nu)}{2\Delta u}\Big). \tag{2.9}$$

Two interesting and useful theorems in [39] (Theorem 3.10 and Theorem 3.11) extend our understanding of the exponential mechanism. First, the sampling distribution of Eq. 2.9 is equivalent to a Laplace mechanism with half of the privacy budget — the normalization constant accounts for the remaining half. Second, for a finite output space $Range(\mathcal{M})$, the probability of sampling values with near optimal utility can be lower bounded. Specifically, for maximum utility $\mathrm{OPT}_u(D)$, size of output space $|\mathcal{R}|$, and size of subset of output space that achieves maximum utility $|\mathcal{R}_{OPT}|$:

$$\Pr\Big[u(D,\nu) \leq \mathrm{OPT}_u(D) - \frac{2\Delta u}{\alpha}\Big(t + \ln\frac{|\mathcal{R}|}{|\mathcal{R}_{OPT}|}\Big)\Big] \leq e^{-t}. \tag{2.10}$$

Functional mechanism Query-answering is not the sole province of differential privacy. Other modeling methods can also reveal the presence of an individual in the data set, and therefore benefit from a private version of their standard procedure. One such example is linear regression. Rather than applying noise to regression outputs, or even to regression parameters, [141] proposes to add Laplace noise to the coefficients of a polynomial representation of the loss function, available for all continuous and differentiable functions as a result of the Stone-Weierstrass theorem. This is noted as being less restrictive than [25], which offers similar work on objective perturbation and requires a convex loss function.

Stochastic gradient descent Just as the functional mechanism exists in settings outside of the classic query-answering setting, so too does the differentially private version of stochastic gradient descent. Proposed in [1], this approach proposes to use differentially private noise in a deep learning optimizer, using the Gaussian noise mechanism, and using clipped gradient values to establish the sensitivity.

2.3.4 Sensitivity calculations

A handful of variations exist to compute sensitivity, sometimes in less-restrictive or empirical settings.

Local sensitivity In many settings, the theoretical sensitivity vastly overestimates the variability of the query outputs on "typical" data. In such cases, the large sensitivity leads to noise that overpowers the data, causing the private output to be useless. The local sensitivity is intended to reflect typical variation in output for query f by measuring the sensitivity with respect to a specific data set D, *restricted to the neighbors of* D. Formally,

$$LS_f(D) = \max_{D' \in neigh(D)} |f(D) - f(D')|. \tag{2.11}$$

Unfortunately, local sensitivity can vary substantially and can be disclosive. In one adaptation called Propose-Test-Release [37], a proposed sensitivity (larger than the local sensitivity but smaller than the global sensitivity) is used, but conditional on passing a test that checks whether high sensitivity data sets are sufficiently far from the given data set.

Smooth sensitivity Smooth sensitivity [103] resolves the high variability of local sensitivity by computing for a particular data set D, the smallest, smooth upper bound on local sensitivities. A smooth sensitivity $S : \mathcal{X} \to \mathbb{R}+$, with $\beta > 0$, satisfies the following requirements:

$$\forall D \in \mathcal{X}: \qquad S(D) \geq LS_f(x) \qquad \text{(upper bound)} \tag{2.12}$$

$$\forall D, D' \in \mathcal{X}, \mathrm{d}_H(D, D') = 1: \qquad S(D) \leq e^{\beta} \cdot S(D'). \quad \text{(smooth)} \tag{2.13}$$

$$\tag{2.14}$$

The smooth sensitivity is defined as the maximum among local sensitivities of other data sets, weighted by their distance to D. A sensitivity is β-*smooth* when defined as follows,

$$S^*_{f,\beta}(D) = \max_{D' \in \mathcal{X}} \left(LS_f(D') \cdot e^{-\beta \, \mathrm{d}_H(D,D')} \right). \tag{2.15}$$

While [103] presents approximation algorithms for computing smooth sensitivities for the median, minimum, cost of a minimum spanning tree, and number of triangles in a graph, it also acknowledges that even approximations can be too difficult in some settings.

Sample and aggregate The sample and aggregate method partitions a data set, computes a query on each subset, then aggregates the responses from each partition in a differentially private way. The partition ensures that the change of any individual point affects only the outcome of the query over that subset, and query outcomes on other subsets remain unchanged. Intuitively, this produces more stable, lower sensitivity, query results. Figure 2.2 from [103] illustrates this procedure. Such a model works when a query can be well approximated by a subset – typically where a single clustering is obvious. While the authors of [103] propose to use this method followed by their smooth sensitivity, the procedure can be used in a generic way, as an extension to any query. In [109] for example, the proposed method uses the sample and aggregate method for its teacher model, where each teacher makes a prediction, the results form a histogram, the histogram counts are perturbed with differentially private noise, then the mode of the noisy histogram is selected. The type of aggregation is largely up to the user, though must be chosen in a manner that is independent of the data [39]. Ideally, the aggregation provides a low and easily-measured sensitivity. Examples include the α-trimmed mean, the Winsorized mean, the median, and *center of attention* [103].

Sensitivity sampling Sensitivity sampling [115] proposes a probabilistic approach to choosing sensitivity based on empirical calculations from repeated samples. In the procedure, the sensitivity is repeatedly computed, forming an

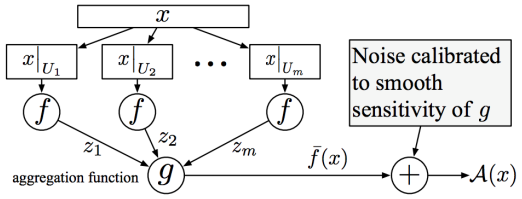


Figure 2.2: Data x is partitioned into m subsets. Query f is performed on each subset producing outputs $z_1, \ldots, z_m$. Outputs are aggregated using function g, and differentially private noise is added based on the sensitivity of the aggregation.

empirical cumulative distribution function (eCDF). A particular level of random differential privacy is then achieved, where a sensitivity corresponding to a higher eCDF value produces the stated privacy level with higher confidence.

2.4 Differentially private data release

While sequential queries face the challenge of loss accumulation, and parallel queries each under-utilize the available data, these interactive settings retain the advantage of knowing the query. They are designed with the minimum noise required *for that query*. In non-interactive settings (also known as data[base/set] release, data synthesis, and synthetic data generation), a single privatizing step produces a synthetic set that aims to be useful for a large class of queries, including those unknown to the data owner. The amount of noise added is typically higher, making utility harder to achieve [39, 38, 27].

For a review of differentially private data release methods, see [22]. The following notes relay a general framework for the family of methods presented there. Among non-parametric methods, Laplace-noised histograms and contingency tables apply in many settings with categorical data, or data that can be easily discretized. A smoothed histogram approach from [136] suggests a linear combination between true and uniform histograms, and another approach there suggests sampling the space of cumulative distribution functions using the exponential mechanism. A method from [55] proposes to add a Gaussian process to a kernel density estimator. Among the parametric methods,

[2] and [93] demonstrate how Multinomial-Dirichlet and Beta-Binomial models, respectively, can be differentially private with specifically crafted priors. Another Bayesian model is proposed in [83], which models a posterior, then repeatedly adds noise to the posterior's sufficient statistics, and samples from each noisy version of the posterior predictive distribution.

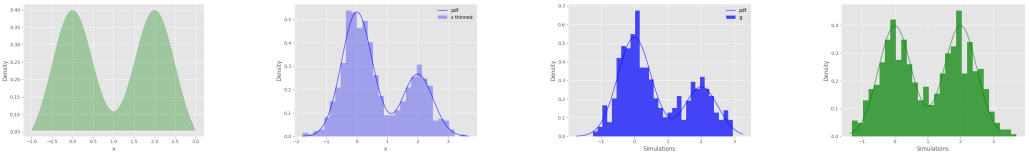
Chapter 3

Importance Weighted Generative Networks

3.1 Introduction

Deep generative models have important applications in many fields: we can automatically generate illustrations for text [140]; simulate video streams [132] or molecular fingerprints [66]; and create privacy-preserving versions of medical time-series data [41]. Such models use a neural network to parameterize a function $G(Z)$, which maps random noise Z to a target probability distribution $\mathbb{P}$. This is achieved by minimizing a loss function between simulations and data, which is equivalent to learning a distribution over simulations that is indistinguishable from $\mathbb{P}$ under an appropriate two-sample test. In this paper we focus on Generative Adversarial Networks (GANs) [50, 6, 18, 76], which incorporate an adversarially learned neural network in the loss function; however the results are also applicable to non-adversarial networks [40, 81].

An interesting challenge arises when we do not have direct access to i.i.d. samples from $\mathbb{P}$. This could arise either because observations are obtained via a biased sampling mechanism [21, 143], or in a transfer learning setting where our target distribution differs from our training distribution. As an example of the former, a dataset of faces generated as part of a university project may contain disproportionately many young adult faces relative to the population. As an example of the latter, a Canadian hospital system might want to customize simulations to its population while still leveraging a training set of patients from the United States (which has a different statistical distribution of medical records). In both cases, and more generally, we want to generate data from a target distribution $\mathbb{P}$ but only have access to representative samples from a *modified* distribution $M\mathbb{P}$. We give a pictorial example of this setting in Figure 3.1.



(a) Target distribution $\mathbb{P}$

(b) Observed distribution $M\mathbb{P}$ and samples from $M\mathbb{P}$

(c) Simulations using a standard estimator

(d) Simulations using an importance weighted estimator

Figure 3.1: If our target distribution $\mathbb{P}$ differs from our observed distribution $M\mathbb{P}$, using the standard estimator will replicate $M\mathbb{P}$, while an importance weighted estimator can replicate the target $\mathbb{P}$.

In some cases, we can approach this problem using existing methods. For example, if we can reduce our problem to a conditional data-generating mechanism, we can employ Conditional Generative Adversarial Networks (C-GANs) or related models [97, 105], which enable conditional sampling given one or more latent variables. However, this requires that M can be described on a low-dimensional space, and that we can sample from our target distribution over that latent space. Further, C-GANs rely on a large, labeled dataset of training samples with diversity over the conditioning variable (within each batch), which becomes a challenge when conditioning on a high-dimensional variable. For example, if we wish to modify a distribution over faces with respect to age, gender and hair length, there may be few exemplars of 80-year-old men with long hair with which to learn the corresponding conditional distribution.

In this paper, we propose an alternate approach based on importance sampling [106]. Our method modifies an existing GAN by rescaling the observed data distribution $M\mathbb{P}$ during training, or equivalently by reweighting the contribution of each data point to the loss function. When training a GAN with samples from $M\mathbb{P}$, the standard estimator equally weights the contribution of each point, yielding an estimator of the loss with respect to $M\mathbb{P}$ and corresponding simulations, as shown in Fig. 3.1b and Fig. 3.1c. This is not ideal.

In order to yield the desired estimator with respect to our target distri-

bution $\mathbb{P}$, we modify the estimator by reweighting the loss function evaluation for each sample. When the Radon-Nikodym derivative between the target and observed distributions (aka the modifier function M) is known, we inversely scale each evaluation by that derivative, yielding the finite-sample importance sampling transform on the estimate, which we call the *importance weighted* estimator. This reweighting asymptotically ensures that discrimination, and the corresponding GAN update, occurs with respect to $\mathbb{P}$ instead of $M\mathbb{P}$, as shown in Fig. 3.1a and Fig. 3.1d.

This approach has multiple advantages and extensions. First, if M is known, we can estimate importance weighted losses using robust estimators like the median-of-means estimator, which is crucial for controlling variance in settings where the modifier function M has a large dynamic range. Second, even when the modifier function is only known up to a scaling factor, we can construct an alternative estimator using self-normalized sampling [114, 106] to use this partial information, while still maintaining asymptotic correctness. Finally and importantly, for the common case of an unknown modifier function, we demonstrate techniques for estimating it from partially labeled data.

Our contributions are as follows: 1) We provide a novel application of traditional importance weighting to deep generative models. This has connections to many types of GAN loss functions through the theory of U-statistics. 2) We propose several variants of our importance weighting framework for different practical scenarios. When dealing with particularly difficult functions M, we propose to use robust median-of-means estimation and show that it has similar theoretical guarantees under weaker assumptions, *i.e.* bounded second moment. When M is not known fully (only up to a scaling factor), we propose a self-normalized estimator. 3) We conduct an extensive experimental evaluation of the proposed methods on both synthetic and real-world data sets. This includes estimating M when less than 4% of the data is labeled with user-provided exemplars.

3.1.1 Related Work

Our method aims to generate samples from a distribution $\mathbb{P}$, given access to samples from $M\mathbb{P}$. While to the best of our knowledge this has

not been explicitly addressed in the GAN literature, several approaches have related goals.

Domain adaptation: Our formulation is related to but distinct from the problem of Domain Adaptation (DA). The challenge of DA is, If I train on one distribution and test on another, how do I maximize performance on test data? Critically, the test data is available and extensively used. Instead, our method solves the problem, Given only a training data distribution, how do I generate from arbitrarily modified versions of it? The former uses two data sets – one source and one target – while the latter uses one dataset and accommodates an arbitrary number of targets. The methodologies are inherently different because the information available is different.

Typical approaches to DA involve finding domain-invariant feature representations for both source and target data. Blitzer, Pereira, Ben-David, and Daume [19, 14, 33] write extensively on techniques involving feature correlation and mutual information within classification settings. Pan, Huang, and Gong [107, 108, 63, 49] propose methods with similar goals that find kernel representations under which source and target distributions are close. The work of [63] and [123] address covariate shift using kernel-based and importance-weighted techniques, but still inhabit a different setting from our problem since they perform estimation on specific source and target data sets.

Recently, the term DA has been used in the context of adversarially-trained image-to-image translation and downstream transfer learning tasks [64, 129, 145, 60]. Typically the goal is to produce representations of the same image in both source and target domains. Such problems begin with data sets from both domains, whereas our setting presents only one source dataset and seeks to generate samples from a hypothetical, user-described target domain.

Inverse probability weighting: Inverse probability weighting (IPW), originally proposed by [62] and still in wide use in the field of survey statistics [92], can be seen as a special case of importance sampling. IPW is a weighting scheme used to correct for biased treatment assignment methods in survey sampling. In such settings, the target distribution is known and the sampling distribution is typically finite and discrete, and can easily be estimated from data.

Conditional GANs: Conditional GANs (C-GANs) are an extension of GANs that aim to simulate from a conditional distribution, given some covariate. In the case where our modifier function M can be represented in terms of a low-dimensional covariate space, and if we can generate samples from the marginal distribution of $M\mathbb{P}$ on that space, then we can, in theory, use a C-GAN to generate samples from $\mathbb{P}$, by conditioning on the sampled covariates. This strategy suffers from two limitations. First, it assumes we can express M in terms of a sampleable distribution on a low-dimensional covariate space. For settings where M varies across many data dimensions or across a high-dimensional latent embedding, this ability to sample becomes untenable. Second, learning a family of conditional distributions is typically more difficult than learning a single joint distribution. As we show in our experiments, C-GANs often fail if there are too few real exemplars for a given covariate setting.

Related to C-GANs, [32] proposes conditional generation and a classifier for assigning samples to specific discriminators. While not mentioned, such a structure could feasibly be used to preferentially sample certain modes, if a correspondence between latent features and numbered modes were known.

Weighted loss: In the context of domain adaptation for data with discrete class labels, the strategy of reweighting the Maximum Mean Discrepancy (MMD) [51] based on class probabilities has been proposed by [139]. This approach, however, differs from ours in several ways: It is limited to class imbalance problems, as opposed to changes in continuous-valued latent features; it requires access to the non-conforming target dataset; it provides no theoretical guarantees about the weighted estimator; and it is not in the generative model setting.

Other uses of importance weights in GANs: The language and use of importance weights is not unique to this application, and has been used for other purposes within the GAN context. In [57], for example, importance weights are used to provide policy gradients for GANs in a discrete-data setting. Our application is different in that our target distribution is not that of our data, as it is in [57]. Instead we view our data as having been modified, and use importance weights to simulate closer to the hypothetical and desired *unmodified* distribution.

3.2 Problem Formulation and Technical Approach

The problem: Given training samples from a distribution $M\mathbb{P}$, our goal is to construct (train) a generator function $G(\cdot)$ that produces i.i.d. samples from a distribution $\mathbb{P}$.

To train $G(\cdot)$, we follow the methodology of a Generative Adversarial Network (GAN) [50]. In brief, a GAN consists of a pair of interacting and evolving neural networks – a generator neural network with outputs that approximate the desired distribution, and a discriminator neural network that distinguishes between increasingly realistic outputs from the generator and samples from a training dataset.

The loss function is a critical feature of the GAN discriminator, and evaluates the closeness between the samples of the generator and those of the training data. Designing good loss functions remains an active area of research [6, 76]. One popular loss function is the Maximum Mean Discrepancy (MMD) [51], a distributional distance that is zero if and only if the two distributions are the same. As such, MMD can be used to prevent mode collapse [117, 28] during training.

Our approach: We are able to train a GAN to generate samples from $\mathbb{P}$ using a simple reweighting modification to the MMD loss function. Reweighting forces the loss function to apply greater penalties in areas of the support where the target and observed distributions differ most.

Below, we formally describe the MMD loss function, and describe its importance weighted variants.

Remark 1 (Extension to other losses). While this paper focuses on the MMD loss, we note that the above estimators can be extended to any estimator that can be expressed as the expectation of some function with respect to one or more distributions. This class includes losses such as squared mean difference between two distributions, cross entropy loss, and autoencoder losses [127, 58, 98]. Such losses can be estimated from data using a combination of U-statistics, V-statistics and sample averages. Each of these statistics can be reweighted, in a manner analogous to the treatment described above. We provide more comprehensive details in Table 3.1, and in Section 3.3.1 we evaluate all three

importance weighting techniques as applied to the standard cross entropy GAN objective.

3.2.1 Maximum Mean Discrepancy between Two Distributions

The MMD projects two distributions $\mathbb{P}$ and $\mathbb{Q}$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, and evaluates the maximum mean distance between the two projections, *i.e.*

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f\in\mathcal{H}} \left(\mathbf{E}_{X\sim\mathbb{P}}[f(X)] - \mathbf{E}_{Y\sim\mathbb{Q}}[f(Y)]\right).$$

If we specify the *kernel mean embedding* $\mu_{\mathbb{P}}$ of $\mathbb{P}$ as $\mu_{\mathbb{P}} = \int k(x,\cdot)d\mathbb{P}(x)$, where $k(\cdot,\cdot)$ is the characteristic kernel defining the RKHS, then we can write the square of this distance as

$$\begin{aligned}
&\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = ||\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}||^2_{\mathcal{H}} \\
&= \mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')] + \mathbb{E}_{Y,Y'\sim\mathbb{Q}}[k(Y,Y')] \\
&- 2\mathbb{E}_{X\sim\mathbb{P},Y\sim\mathbb{Q}}[k(X,Y)]. \qquad (3.1)
\end{aligned}$$

In order to be a useful loss function for training a neural network, we must be able to estimate $\mathrm{MMD}^2(\mathbb{P},\mathbb{Q})$ from data, and compute gradients of this estimate with respect to the network parameters. Let $\{x_i\}_n$ be a sample $\{X_1 = x_1,\ldots,X_n = x_n\} : X_i \sim \mathbb{P}$, and $\{y_i\}_m$ be a sample $\{Y_1 = y_1,\ldots,Y_m = y_m\} : Y_i \sim \mathbb{Q}$. We can construct an unbiased estimator $\widehat{\mathrm{MMD}}^2(\mathbb{P},\mathbb{Q})$ of $\mathrm{MMD}^2(\mathbb{P},\mathbb{Q})$ [51] using these samples as

$$\begin{aligned}
\widehat{\mathrm{MMD}}^2(\mathbb{P},\mathbb{Q}) &= \tfrac{1}{n(n-1)}\textstyle\sum_{i\neq j}^{n} k(x_i,x_j) \\
&+ \tfrac{1}{m(m-1)}\textstyle\sum_{i\neq j}^{m} k(y_i,y_j) \\
&- \tfrac{2}{nm}\textstyle\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i,y_j). \qquad (3.2)
\end{aligned}$$

3.2.2 Importance Weighted Estimator for Known M

We begin with the case where M (which relates the distribution of the samples and the desired distribution; formally the Radon-Nikodym derivative)

is known. Here, the reweighting of our loss function can be framed as an *importance sampling* problem: we want to estimate $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q})$, which is in terms of the target distribution $\mathbb{P}$ and the distribution $\mathbb{Q}$ implied by our generator, but we have samples from the modified $M\mathbb{P}$. Importance sampling [106] provides a method for constructing an estimator for the expectation of a function $\phi(X)$ with respect to a distribution $\mathbb{P}$, by taking an appropriately weighted sum of evaluations of ϕ at values sampled from a different distribution. We can therefore modify the estimator in (3.2) by weighting each term in the estimator involving data point x_i using the likelihood ratio $\mathbb{P}(x_i)/M(x_i)\mathbb{P}(x_i) = 1/M(x_i)$, yielding an unbiased importance weighted estimator that takes the form

$$\begin{aligned}\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q}) = &\tfrac{1}{n(n-1)} \textstyle\sum_{i \neq j}^{n} \frac{k(x_i, x_j)}{M(x_i)M(x_j)} \\ &+ \tfrac{1}{m(m-1)} \textstyle\sum_{i \neq j}^{m} k(y_i, y_j) \\ &- \tfrac{2}{nm} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{k(x_i, y_j)}{M(x_i)}. \end{aligned} \tag{3.3}$$

While importance weighting using the likelihood ratio yields an unbiased estimator (3.3), the estimator may not concentrate well because the weights $\{1/M(x_i)\}_n$ may be large or even unbounded. We now provide a concentration bound for the estimator in (3.3) for the case where weights $\{1/M(x_i)\}_n$ are upper-bounded by some maximum value.

Theorem 1. Let $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})$ be the unbiased, importance weighted estimator for $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q})$ defined in (3.3), given m i.i.d samples from $M\mathbb{P}$ and $\mathbb{Q}$, and maximum kernel value K. Further assume that $1 \leq 1/M(x) \leq W$ for all $x \in \mathcal{X}$. Then

$$\begin{aligned}&\mathbb{P}\left(\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q}) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) > t\right) \leq C, \\ &\text{where } C = \exp((-2t^2 m_2)/(K^2(W+1)^4)) \\ &\qquad m_2 := \lfloor \tfrac{m}{2} \rfloor\end{aligned}$$

These guarantees are based on estimator guarantees in [51], which in turn build on classical results by Hoeffding [59, 58]. We defer the proof of this theorem to Appendix A.1.

3.2.3 Robust Importance Weighted Estimator for Known M

Theorem 1 is sufficient to guarantee good concentration of our importance weighted estimator only when $1/M(x)$ is uniformly bounded by some constant W, which is not too large. Many class imbalance problems fall into this setting. However, $1/M(x)$ may be unbounded in practice. Therefore, we now introduce a different estimator, which enjoys good concentration even when only $\mathbb{E}_{X\sim M\mathbb{P}}[1/M(X)^2]$ is bounded, while $1/M(x)$ may be unbounded for many values of x.

The estimator is based on the classical idea of median of means [100, 65, 4, 74][1]. Given m samples from $M\mathbb{P}$ and $\mathbb{Q}$, we divide these samples uniformly at random into k equal sized groups, indexed $\{(1), ..., (k)\}$. Let $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(i)}$ be the value obtained when the estimator in (3.3) is applied on the i-th group of samples. Then our median of means based estimator is given by

$$\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P}, \mathbb{Q}) = \mathrm{median}\left\{\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(1)}, \ldots, \widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(k)}\right\}. \tag{3.4}$$

Theorem 2. Let $\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P}, \mathbb{Q})$ be the asymptotically unbiased median of means estimator defined in (3.4) using $k = mt^2/(8K^2\sigma^2)$ groups. Further assume that $n{=}m$ and let $W_2 = \mathbb{E}_{X\sim M\mathbb{P}}[\frac{1}{M(X)^2}]$ be bounded. Then

$$\begin{aligned}
&\mathbb{P}\left(|\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P}, \mathbb{Q}) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q})| > t\right) \leq C,\\
&\text{where } C = \exp((-mt^2)/(64K^2\sigma^2))\\
&\qquad \sigma^2 = O\left(W_2^2 + \mathrm{MMD}^4(\mathbb{P}, \mathbb{Q})\right).
\end{aligned}$$

We defer the proof of this theorem to Section A.2. Note that the confidence bound in Theorem 2 depends on the term W_2 being bounded. This is the second moment of $1/M(X)$ where $X \sim M\mathbb{P}$. Thus, unlike in Theorem 1, this confidence bound may still hold even if $1/M(x)$ is *not uniformly bounded*. When $1/M(X)$ is heavy-tailed with finite variance, *e.g.* Pareto ($\alpha > 2$) or log-normal, then Theorem 2 is valid but Theorem 1 does not apply.

[1] [74] appeared concurrently and contains a different approach for the unweighted estimator. Comparisons are left for future work.

In addition to increased robustness, the median of means MMD estimator is more computationally efficient: since calculating $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})$ scales quadratically in the batch size, using the median of means estimator introduces a speed-up that is linear in the number of groups.

3.2.4 Self-normalized Importance Weights for Unknown M

To specify M, we must know the forms of our target and observed distributions along any marginals where the two differ. In some settings this is available: consider for example a class rebalancing setting where we have class labels and a desired class ratio, and can estimate the observed class ratio from data. This, however, may be infeasible if M is continuous and/or varies over several dimensions, particularly if data are arriving in a streaming manner. In such a setting it may be easier to specify a *thinning function* T *that is proportional to* M*, i.e.* $M\mathbb{P} = \frac{T\mathbb{P}}{Z}$ for some unknown Z, than to estimate M directly. This is because T can be directly obtained from an estimate of how much a given location is underestimated, without any knowledge of the underlying distribution.

This setting—where the $1/M$ weights used in Section 3.2.2 are only known up to a normalizing constant—motivates the use of a *self-normalized* importance sampling scheme, where the weights $w_i \propto \frac{\mathbb{P}(x_i)}{M(x_i)\mathbb{P}(x_i)} = \frac{Z}{T(x_i)}$ are normalized to sum to one [114, 106]. For example, by letting $w_i = \frac{1}{T(x_i)}$, the resulting self-normalized estimator for the squared MMD takes the form

$$\begin{aligned}\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q}) = &\frac{\sum_{i\neq j}^{n} w_i w_j k(x_i, x_j)}{\sum_{i\neq j}^{n} w_i w_j} \\ &+ \sum_{i\neq j}^{m} \frac{k(y_i, y_j)}{m(m-1)} \\ &- 2\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} w_i k(x_i, y_j)}{m\sum_{i=1}^{n} w_i}. \end{aligned} \tag{3.5}$$

While use of self-normalized weights means this self-normalized estimator is biased, it is asymptotically unbiased, with the bias decreasing at a rate of $1/n$ [68]. Although we have motivated self-normalized weights out of necessity, in practice they often trade off bias for reduced variance, making them preferable in some practical applications [106].

More generally, in addition to not knowing the normalizing constant Z, we might also not know the thinning function T. For example, T might vary

along some latent dimension—perhaps we want to have more images of people fitting a certain aesthetic, rather than corresponding to a certain observed covariate or class. In this setting, a practitioner may be able to estimate $T(x_i)$, or equivalently w_i, for a small number of training points x_i, by considering how much those training points are under- or over-represented. Continuous-valued latent preferences can therefore be expressed by applying higher weights to points deemed more appealing. From here, we can use function estimation techniques, such as neural network regression, to estimate T from a small number of labeled data points.

3.2.5 Approximate Importance Weighting by Data Duplication

In the importance weighting scheme described above, each data point is assigned a weight $1/M(x_i)$. We can obtain an approximation to this method by including $\lceil 1/M(x_i) \rceil$ duplicates of data point x_i in our training set. We refer to this approach as *importance duplication*. Importance duplication obviously introduces discretization errors, and if our estimator is a U-statistic it will introduce bias (*e.g.* in the MMD example, if two or more copies of the data point x_i appear in a minibatch, then $k(x_i, x_i)$ will appear in the first term of (3.2)). However, as we show in the experimental setting, even though this approach lacks theoretical guarantees it provides generally good performance.

Data duplication can be done as a pre-processing step, making it an appealing choice if we have an existing GAN implementation that we do not wish to modify. In other settings, it is less appealing, since duplicating data adds an additional step and increases the amount of data the algorithm must process. Further, if we were to use this approximation in a setting where M is unknown, we would have to perform this data duplication on the fly as our estimate of M changes.

3.3 Evaluation

In this section, we show that our estimators, in conjunction with an appropriate generator network, allow us to generate simulations that are close in distribution to our target distribution, even when we only have access to

Table 3.1: Constructing importance weighted estimators for losses involving U-statistics, V-statistics and sample averages. Here, $\mathcal{U}$ is the set of all r-tuples of numbers from 1 to n without repeats, and $\mathcal{V}$ is the set of r-tuples allowing repeats. Below, let $X_{u,*} = X_{u_1}, \ldots, X_{u_r}$.

	$\widehat{\mathrm{D}}(\mathbb{P}, \mathbb{Q})$	$\widehat{\mathrm{D}}_{IW}(\mathbb{P}, \mathbb{Q})$	$\widehat{\mathrm{D}}_{SNIW}(\mathbb{P}, \mathbb{Q})$
U-statistic	$\frac{1}{{}^nP_r} \sum_{u \in \mathcal{U}} g(X_{u,*})$	$\frac{1}{{}^nP_r} \sum_{u \in \mathcal{U}} \frac{g(X_{u,*})}{M(X_{u_1}) \cdots M(X_{u_r})}$	$\frac{\sum_{u \in \mathcal{U}} w_{u_1} \cdots w_{u_r} g(X_{u,*})}{\sum_{u \in \mathcal{U}} w_{u_1} \cdots w_{u_r}}$
V-statistic	$\frac{1}{n^r} \sum_{v \in \mathcal{V}} g(X_{v,*})$	$\frac{1}{n^r} \sum_{v \in \mathcal{V}} \frac{g(X_{v,*})}{M(X_{v_1}) \cdots M(X_{v_r})}$	$\frac{\sum_{v \in \mathcal{V}} w_{v_1} \cdots w_{v_r} g(X_{v,*})}{\sum_{v_r=1}^n w_{v_1} \cdots w_{v_r}}$
Average	$\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m f(X_i, Y_j)$	$\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \frac{f(X_i, Y_j)}{M(X_i)}$	$\frac{\sum_{i=1}^n w_i \sum_{j=1}^m f(X_i, Y_j)}{m \sum_{i=1}^n w_i}$

this distribution via a biased sampling mechanism. Further, we show that our method performs comparably with, or better than, conditional GAN baselines.

Most of our weighted GAN models are based on the MMD-GAN of [76], replacing the original MMD loss with either our importance weighted loss $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})$ (IW-MMD), our median of means loss $\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P}, \mathbb{Q})$ (MIW-MMD), or our self-normalized loss $\widehat{\mathrm{MMD}}^2_{SNIW}(\mathbb{P}, \mathbb{Q})$ (SNIW-MMD). We also use a standard MMD loss with an importance duplicated dataset (ID-MMD). Other losses used in [76] are also appropriately weighted, following the form in Table 3.1. In the synthetic data examples of Section 3.3.1, the kernel is a fixed radial basis function, while in all other sections it is adversarially trained using a discriminator network as in [76].

To demonstrate that our method is applicable to other losses, in Section 3.3.1 we also create models that use the standard cross entropy GAN loss, replacing this loss with either an importance weighted estimator (IW-CE), a median of means estimator (MIW-CE) or a self-normalized estimator (SNIW-CE). We also combine a standard cross entropy loss with an importance duplicated dataset (ID-CE). These models used a two-layer feedforward neural network with ten nodes per layer.

Where appropriate, we compare against a conditional GAN (C-GAN). If M is known exactly and expressible in terms of a lower-dimensional co-

variate space, a conditional GAN (C-GAN) offers an alternative method to sample from $\mathbb{P}$: learn the appropriate conditional distributions given each covariate value, sample new covariate values, and then sample from $\mathbb{P}$ using each conditional distribution.

3.3.1 Can GANs with Importance Weighted Estimators Recover Target Distributions, Given M?

To evaluate whether using importance weighted estimators can recover target distributions, we consider a synthetically generated distribution that has been manipulated along a latent dimension. Under the target distribution, a latent representation θ_i of each data point lives in a ten-dimensional space, with each dimension independently Uniform(0,1). The observed data points x_i are then obtained as $\theta_i^T F$, where $F_{ij} \sim \mathcal{N}(0, 1)$ represents a fixed mapping between the latent space and D-dimensional observation space. In the training data, the first dimension of θ_i has distribution $p(\theta) = 2\theta, 0 < \theta \leq 1$. We assume that the modifying function $M(x_i) = 2\theta_{i,1}$ is observed, but that the remaining latent dimensions are unobserved.

In our experiments, we generate samples from the target distribution using each of the methods described above, and include weighted versions of the cross entropy GAN to demonstrate that importance weighting can be generalized to other losses.

To compare methods, we report the empirically estimated KL divergence between the target and generated samples in Table 3.2. Similar results using squared MMD and energy distance are shown in Table A.1 and Table A.2 in Appendix A.3. For varying real dimensions D, importance weighted methods outperform C-GAN under a variety of measures.

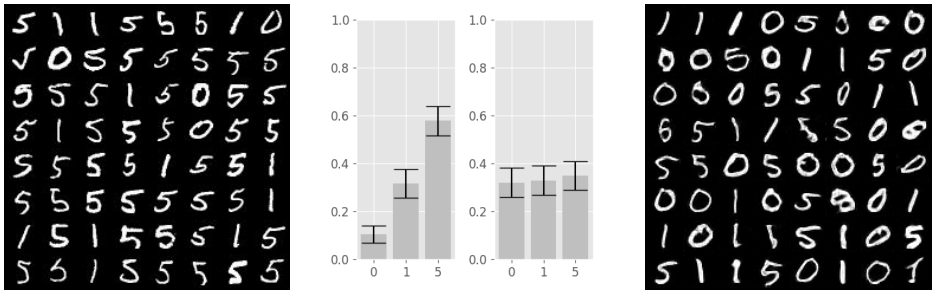
In some instances C-GAN performs well in two dimensions, but deteriorates quickly as the problem becomes more challenging with higher dimensions. We also note that many runs of C-GAN either ran into numerical issues or diverged; in these cases we report the best score among runs, before training failure.

While the above experiment can be evaluated numerically and provide good results for thinning on a continuous-valued variable, it is difficult to visu-

Table 3.2: Estimated KL divergence between generated and target samples (mean ± standard deviation over 20 runs).

Model	2D	4D	10D
IW-CE	0.1768 ± 0.0635	0.4934 ± 0.1238	2.7945 ± 0.5966
MIW-CE	0.3265 ± 0.1071	0.6251 ± 0.1343	3.3093 ± 0.7179
SNIW-CE	0.0925 ± 0.0272	0.3864 ± 0.1478	2.3060 ± 0.6915
ID-CE	0.1526 ± 0.0332	0.3444 ± 0.0766	1.4128 ± 0.3288
IW-MMD	**0.0343 ± 0.0230**	**0.0037 ± 0.0489**	**0.5133 ± 0.1718**
MIW-MMD	0.2698 ± 0.0618	0.0939 ± 0.0522	0.8501 ± 0.3271
SNIW-MMD	0.0451 ± 0.0132	0.1435 ± 0.0377	0.6623 ± 0.0918
C-GAN	0.0879 ± 0.0405	0.3108 ± 0.0982	6.9016 ± 2.8406

alize the outcome. In order to better visualize whether the target distribution is correctly achieved, we also run experiments with explicit and easily measurable class distributions. In Figure 3.2, we show a class rebalancing problem on MNIST digits, where an initial uneven distribution between three classes can be accurately rebalanced. We also show good performance modifying a balanced distribution to specific boosted levels (see Appendix A.3). Together, these experiments provide evidence that importance weighting controls the simulated distribution in the desired way.

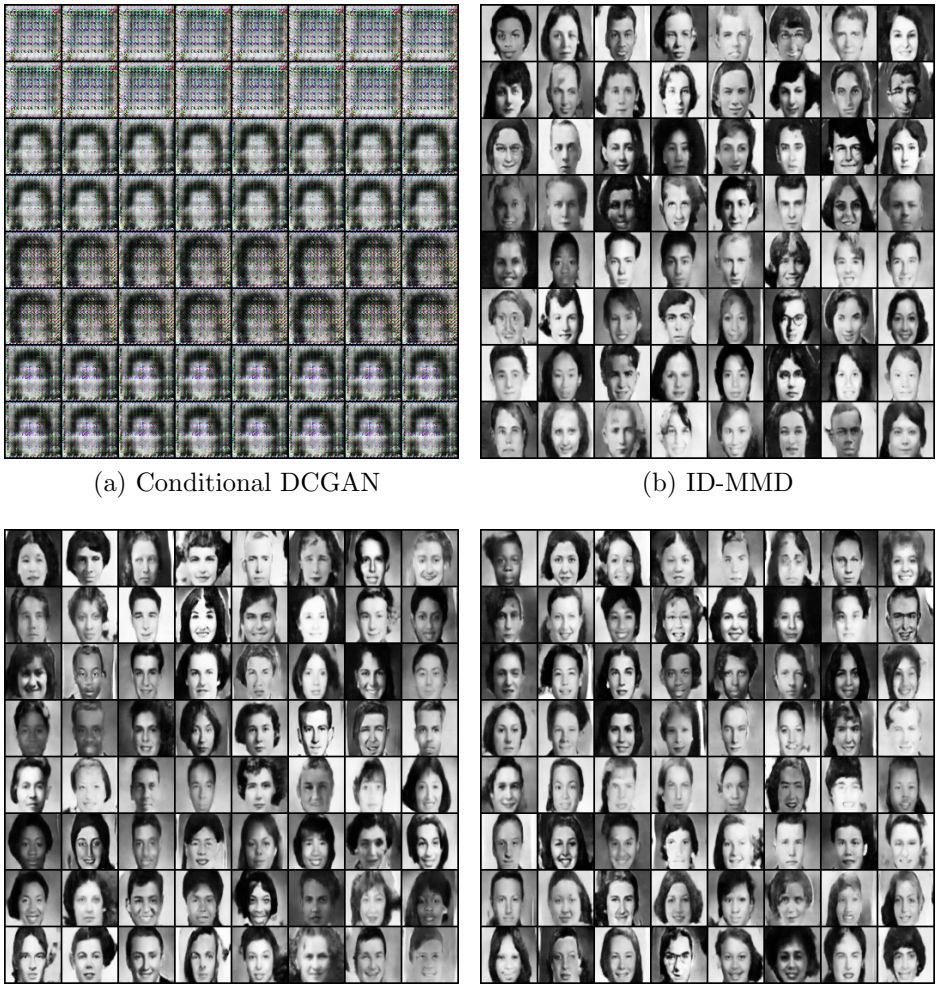


(a) Source, uneven distribution of 0s, 1s, and 5s

(b) Source (left), simulation (right); target of $\frac{1}{3}$-$\frac{1}{3}$-$\frac{1}{3}$

(c) Simulations, balanced distribution

Figure 3.2: Importance weights are used to accurately rebalance an uneven class distribution.



(a) Conditional DCGAN (b) ID-MMD

(c) Importance Weighting (IW-MMD) (d) Median of Means (MIW-MMD)

Figure 3.3: Example generated images for all example networks, Yearbook dataset [48]. Target distribution is uniform across half-decades, while the training set is unbalanced.

3.3.2 In a High-dimensional Image Setting, How Does Importance Weighting Compare with Conditional Generation?

Next we evaluate performance of importance weighted MMD on high-dimensional image generation. In this section we address two questions: Can our estimators generate simulations from $\mathbb{P}$ in such a setting, and how do

the resulting images compare with those obtained using a C-GAN? To do so, we evaluate several generative models on the Yearbook dataset [48], which contains over 37,000 high school yearbook photos across over 100 years and demonstrates evolving styles and demographics. The goal is to produce images uniformly across each half decade. Each GAN, however, is trained on the original dataset, which contains many more photos from recent decades.

Since we have specified M in terms of a single covariate (time), we can compare with C-GANs. For the C-GAN, we use a conditional version of the standard DCGAN architecture (C-DCGAN) [110].

Figure 3.3 shows generated images from each network. All networks were trained until convergence. The images show a diversity across hairstyles, demographics and facial expressions, indicating the successful temporal rebalancing. Even while importance duplication introduces approximations and lacks the theoretical guarantees of the other two methods, all three importance-based methods achieve comparable quality. Since some covariates have fewer than 65 images, C-DCGAN cannot learn the conditional distributions, and is unstable across a variety of training parameters. Implementation details and additional experiments are shown in Appendix A.3.

3.3.3 When M Is Unknown, but Can Be Estimated Up to a Normalizing Constant on a Subset of Data, Are We Able to Sample from our Target Distribution?

In many settings, especially those with high-dimensional latent features, we will not know the functional form of M, or even the corresponding thinning function T. We would still, however, like to be able to express a preference for certain areas of the latent space. To do so, we propose labeling a small subset of data using weights that correspond to preference. To expand those weights to the entire dataset, we train a neural network called the estimated weighting function. This weighting function takes encoded images as input, and outputs continuous-valued weights. Since this function exists in a high-dimensional space that changes as the encoder is updated, and since we do not know the full observed distribution on this space, we are in a setting unsuitable for conditional methods, and therefore use self-normalized estimators (SNIW-MMD).

We evaluate using a collection of sevens from the MNIST dataset, where the goal is to generate more European-style sevens with horizontal bars. Out of 5915 images, 200 were manually labeled with a weight (reciprocal of a thinning function value), where sevens with no horizontal bar were assigned a 1, and sevens with horizontal bars were assigned weights between 2 and 9 based on the width of the bar.

(a) Data (b) Generator (c) KS distance

Figure 3.4: Partial labeling and an importance weighted estimator boost the presence of sevens with horizontal bars. In 3.4a and 3.4b, samples are sorted by predicted weight, and in 3.4c, the empirical CDFs of data, generated, and importance duplicated draws, are shown, where the latter serves as a theoretical target. The generated distribution is close in distance to the target.

Fig. 3.4a shows 64 real images, sorted in terms of their predicted weights – note that the majority have no horizontal bar. Fig. 3.4b shows 64 generated simulations, sorted in the same manner, clearly showing an increase in the number of horizontal-bar sevens.

To test the quantitative performance, we display and compare the empirical CDFs of weights from simulations, data, and importance duplicated data. For example, if a batch of data $[A, B, C]$ has weights $[1, 3, 2]$, this implies that we expected three times as many B-like points and two times as many C-like points as A-like points. A simulator that achieves this target produces simulations like $[A, B, B, B, C, C]$ with weights $[1, 3, 3, 3, 2, 2]$, equivalent to an importance duplication of data weights. Using importance duplicated weights as a theoretical target, we measure our model's performance by computing

the Kolmogorov-Smirnov (KS) distance between CDFs of simulated and importance duplicated weights. Fig. 3.4c shows a small distributional distance between simulations and their theoretical target, with $d_{KS} = 0.03$, $p = 0.457$.

3.4 Conclusions and Future Work

We present three estimators for the MMD (and a wide class of other loss functions) between target distribution $\mathbb{P}$ and the distribution $\mathbb{Q}$ implied by our generator. These estimators can be used to train a GAN to simulate from the target distribution $\mathbb{P}$, given samples from a modified distribution $M\mathbb{P}$. We present solutions for when M is potentially unbounded, is unknown, or is known only up to a scaling factor.

We demonstrate that importance weighted estimators allow deep generative models to match target distributions for common and challenging cases with continuous-valued, multivariate latent features. This method avoids heuristics while providing good empirical performance and theoretical guarantees.

Though the median of means estimator offers a more robust estimate of the MMD, we may still experience high variance in our estimates, for example if we rarely see data points from a class we want to boost. An interesting future line of research is exploring how variance-reduction techniques [35] or adaptive batch sizes [34] could be used to overcome this problem.

Chapter 4

Synthetic Medical Records

4.1 Problem Statement

Synthetic medical records provide a valuable and challenging data set for applications of generative models and privacy. While global health organizations often want to share data about their populations or treatments, regulation can hinder the dissemination of information for reasons of privacy protection.

In this project, a medical data set is available for statistical study, but cannot be used in publication by law. To address this, a synthetic set is generated instead, and its distributional closeness and privacy levels are studied.

4.2 GAN Pre-Processing

To comply with Chinese policy, we report inference for data generated by a Generative Adversarial Network (GAN, [50]), which replicates the distribution underlying the raw data. GAN is a machine learning algorithm which simultaneously trains a generative model and a discriminative model on a training dataset (in our case, the raw EHR dataset). The generative model simulates the training data distribution in order to trick the discriminative model. Meanwhile, the discriminative model learns to optimally distinguish between data and simulations. During training, the generative model uses gradient information from the discriminative model to produce better simulations. After training, the generative model can be used to generate an arbitrary number of simulations which are similar in distribution to the original dataset. In our case, we generate a simulated dataset of the same size as the raw EHR dataset.

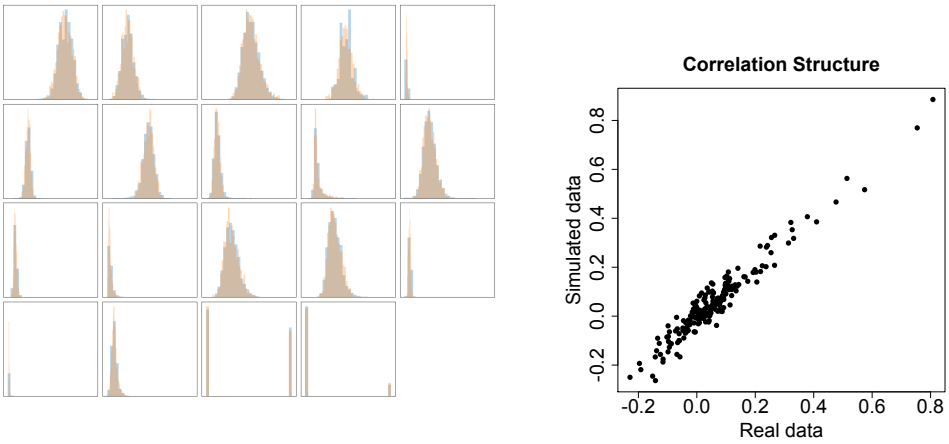
For this application, we train on a dataset where columns of continuous variables are standardized, and corresponding outputs are then re-scaled at simulation time. To accommodate binary variables, we allow the GAN to simulate continuous values, and round corresponding outputs to 0 or 1. We use the architecture of MMD-GAN [76], which uses the maximum mean discrepancy (MMD) [51], a distributional distance, to compare real data and simulations. Our implementation uses encoder and decoder networks each containing three layers of 100 nodes, connected by a bottleneck layer of 64 nodes, and with exponential linear unit activations. In our optimization, we use RMSProp with a learning rate of 0.001, and we weight the MMD in our discriminator loss function by 0.1.

Our model reaches a stable point, where both marginal distributions and pairwise correlations agree with the raw data (see Figure 4.1). Moreover, the classifiers we consider have similar prediction performance on the two data sets. Therefore, we only report the results based on the replicated EHR data (referring to as EHR data hereafter). To the extent to which the preprocessed data set retains all information and structure of the original data any inference other than subject-specific summaries remains practically unchanged.

4.3 Implementation Details

To evaluate the privacy of the simulated set, we measure two types of risk: presence disclosure and attribute disclosure [29]. Presence disclosure is the ability to determine whether a candidate point was included in the training dataset. Attribute disclosure is the ability to predict other attributes of a candidate point, given partial information about that point. For both settings, we choose three sets of equal size – 5% of the training data, a heldout set for testing, and a heldout set for baseline comparison – then estimate the sensitivity and precision of classification schemes.

For presence disclosure, we sample a candidate from the union of training and testing sets, and classify whether the candidate was included in the training set based on the presence of an ϵ-close neighbor in the simulated set. For large ϵ, the notion of ϵ-closeness is not informative, since many points are returned as neighbors, and precision scores hover around 50% – no bet-



(a) Marginal distributions

(b) Pairwise correlations

Figure 4.1: GAN-preprocessed EHR data versus raw EHR data. (a) Marginal distribution of each variable. For each variable, the two overlaid histograms show the agreement between the preprocessed and the raw data. The variable names and ranges are deliberately not shown. (b) Correlation of each pair of variables.

ter than random guessing. For small ϵ, few points are returned as neighbors, and neighbors are more likely to be correctly guessed, since the requirement is for a neighbor to be nearly identical to the candidate point. To reflect the optimal privacy standard, we report scores using the largest ϵ for which precision exceeds 55%. This yields the largest sensitivity under non-trivial risk, where a higher sensitivity indicates greater ability to identify a participant. At $\epsilon = 9.5$, the sensitivity of this classification is 0.0005, indicating that compromised training points would be identifiable only 0.05% of the time.

For attribute disclosure, we sample as above, and classify whether unknown features of a candidate point can be correctly estimated to within 5% of the true value, by averaging each feature over the candidate's five nearest neighbors in the simulated set. We report values for the case in which half of the candidate's features are known, and the other half are imputed, and note that performance did not change significantly when the percentage of known values differed. The sensitivity and precision scores of this classification are

0.31 and 0.72, respectively, indicating that unknown features would be correctly guessed 31% of the time, and features claiming to be within 5% of the true value would be within that range 72% of the time.

We note that privacy and accuracy goals are inherently opposed. An increase in privacy corresponds to a simulated set with less information about individual data points, and vice versa. As a general guideline, we aim to satisfy privacy requirements while preserving (as much as possible) the utility of the simulations. In the specific case of attribute risk, we recognize that scores depend on the correlation structure of the data, where highly correlated features are more susceptible to attribute disclosure. As a baseline, we compared attribute risk scores of simulations to those of the final heldout set, and found that both were approximately 30% and 70%, respectively.

Chapter 5

Support Points and Privacy

5.1 Introduction

Practitioners often want to share data publicly, while also respecting the privacy of individuals. In an effort to preserve privacy, data stewards will anonynimize, redact, summarize, randomize, or add noise to data before publication, in order to make it hard for an adversary to identify individuals. Privacy breaches can lead to financial, political, and personal losses, and they have warranted the attention of large businesses and institutions.

In most discussions, researchers accept a trade-off between privacy and utility. Intuitively, the most private summary of data is a uniform distribution over its support – it is also the least informative, because an adversary cannot identify an individual with better than random chance. The more a distribution differs from the uniform distribution the more informative it is, both in the practical and information-theoretic senses.

The method of differential privacy is particularly appealing because it allows a data steward to make mathematical guarantees about their disclosure, and it is usually offered as a reason for individuals to feel comfortable with contributing their data. From this perspective, privacy is available wherever and to whatever degree it is demanded. As a result, a large class of privacy research has flourished, focusing on ways to induce differential privacy guarantees.

To frame the discussion, consider that the promise of data collection is the ability to learn underlying behavior in a way that generalizes to new data. If the underlying distribution were known, it could be communicated directly without disclosing *any* individual information. Since this is not available in practice, information is derived from finite samples, which are the

best-available expression of the true distribution. A private data set is therefore one that communicates a similar distribution while providing individual privacy protection.

Many privatization schemes are designed for specific queries on the data in order to minimize noise and maximize utility of the result. Many schemes also use counts of data points on a discrete grid (binned data), which can present challenges in naturally-sparse high-dimensional space. This work explores settings with no restrictions on query type (i.e. allowing the identity query), and aims to accommodate high-dimensional, continuous-valued data — in fact, the methods presented accommodate arbitrary data so long as a metric can be defined.

Private data synthesis schemes can take two forms: a standard synthesis model with privatized gradients (e.g. [1]), and a standard synthesis model with a privatized target representation of the data (e.g. [82]). We explore the latter, based on our conceptual framework of communicating distributional information, and given our experience with the method of support points for representing high-dimensional point sets.

Below, Section 5.2 contains background details, Section 5.3 introduces our method for sampling private synthetic point sets, Section 5.4 introduces a computation-saving adaptation for sampling private support points, Section 5.5 describes convergence diagnostics, Section 5.6 proposes methods for expanding private support points to full synthetic data sets, Section 5.7 provides experimental results, Section 5.8 describes and contextualizes related work, and Section 5.9 contains a concluding discussion.

5.2 Background

5.2.1 Differential Privacy

The intuition of differential privacy is to share specific information (such as the mean) of a data set while hiding the presence of individuals. To accomplish this, the information is randomly corrupted with noise from a known distribution. As a result, variability in communicated information can be plausibly attributed to noise, rather than the presence (or absence) of any

individual point.

Consider the standard model of differential privacy as defined by Dwork in [36]. For data set D, deterministic function f, and noise η, a randomized function $\mathcal{M}(D) = f(D) + \eta$ induces a distribution around the true function response. By definition, $\mathcal{M}$ gives α-differential privacy if for all data sets D and D' differing in at most one element ("neighboring sets"), and all $\nu \subseteq Range(\mathcal{M})$,

$$\frac{\Pr[\mathcal{M}(D) \in \nu]}{\Pr[\mathcal{M}(D') \in \nu]} \leq \exp(\alpha), \tag{5.1}$$

i.e. the density ratio for observing any randomized function output is bounded.

Laplace Mechanism In the simplest case known as the *Laplace mechanism*, $\mathcal{M}$ adds Laplace noise to the function response, and the density ratio bound of Eq. 5.1 is derived as follows:

$$\frac{\text{Lap}(f(D), \lambda)}{\text{Lap}(f(D'), \lambda)} \leq \exp\left(\frac{|f(D) - f(D')|}{\lambda}\right) \leq \exp\left(\frac{\Delta f}{\lambda}\right) := \exp(\alpha), \tag{5.2}$$

where Δf is called the *sensitivity*, which represents the largest possible difference in function responses over pairs of neighboring data sets. Here, we use its generalization called the *exponential mechanism*, which incorporates a *score function* when comparing responses from each data set. A brief description is included below, and [95] contains a full review.

Exponential mechanism For query responses in discrete or categorical space, adding continuous numeric noise does not apply. Consider the question "What is the most common name in the database?" There is no clear sense of how to add noise in this setting. Another classic example is privatizing individual bids in an auction, where auction revenue can relate to price in a complex way, and where bids must be credible after privatization. At a certain point, a price increase might drastically change the expected revenue, making it an unreasonable value. In these cases, a utility function acts as a link between the output space and a space that varies more smoothly, in which noise can be added. This procedure is called the exponential mechanism [95].

In the case of the auction, the utility function links the price space to the revenue space, and sampling occurs with respect to the revenue space.

Formally, the exponential mechanism defines a utility function on the joint space of query inputs and outputs, and samples outputs with probability proportional to their utility score. For a utility function $\mu : \mathcal{X} \times \mathcal{M} \to \mathbb{R}$, the sensitivity is defined as

$$\Delta u = \max_{\nu \in \mathcal{M}} \max_{D, D'} |u(D, \nu) - u(D', \nu)|. \tag{5.3}$$

and the sampling distribution is defined as

$$\Pr[D] \propto \exp\Big(\frac{\alpha u(D, \nu)}{2\Delta u}\Big). \tag{5.4}$$

For neighboring D and D', the value in Eq. 5.4 and its normalization constant can change by a maximum factor of $\exp(\alpha/2)$, yielding a density ratio bound of $\exp(\alpha)$ as desired.

One surprisingly straightforward application of the exponential mechanism is the one posterior sample (OPS) mechanism of [135]. The authors show that for a bounded log-likelihood, a single step of Markov Chain Monte Carlo will sample according to a density ratio that can be expressed as an exponential raised to the difference of bounded log-likelihoods — this corresponds exactly to using the exponential mechanism with the log-likelihood as the score function.

5.2.2 Energy Distance

The energy distance [127] is a distributional distance metric evaluated over finite sample sets. It is especially useful when the functional form of a distribution is not known. For two data sets $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_n\}$, the squared energy distance (hereafter "energy") is defined as:

$$e_{N,n}(X, Y) = \frac{2}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} ||x_i - y_j||_p - \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} ||x_i - x_j||_p - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} ||y_i - y_j||_p, \tag{5.5}$$

with $p = 2$. Generalized distance metrics in place of the $||\cdot||_2$ Euclidean norm also hold under certain conditions [88]. For the remainder of this work, let $e(\cdot, \cdot)$ represent the function, and e on its own represent the output of this function on $\mathbb{R}$.

5.2.3 Support Points

In resource-constrained environments or when given highly correlated data, it can be useful to have small summaries of large data sets. This classical problem has spurred a handful of data reduction methods including coresets [3], support points [91], and clustering.

Support points are simple to define, easy to compute, not restricted to being a subset of data (as with coresets), and able to avoid issues of selecting an inappropriate clustering method (e.g. k-means clustering might not be appropriate when clusters have different sizes and variance). Support points for a data set X are defined as

$$Y^* = \underset{y_1,\ldots,y_n}{\arg\min} \; e_{N,n}(X, Y), \tag{5.6}$$

and can be approximated using standard optimization libraries or via the convex surrogate formulation found in [91].

5.3 Sampling Private Synthetic Sets via the Exponential Mechanism with Energy Distance

5.3.1 Motivation and Summary

Given a data set with N points, we might want a private synthetic set with the same distributional information. As described in Section 5.2, differential privacy is satisfied by adding noise to a query response, where noise variance is proportional to the maximum change of the response. When privatizing a full data set, the location of each point can be interpreted as a query, and the maximum that each point can change is the maximum span of the support. For data on $[0, 1]^d$, for example, the sensitivity of each point equals one, and each point would receive noise sampled from $\text{Lap}(1/\alpha)$, for

α-differential privacy. For lower levels of α, this level of noise quickly degrades distribution information.

Alternatively, we can use a procedure that directly incorporates the notion of distributional closeness, in terms of the energy distance. Given the sensitivity of the energy distance, the exponential mechanism provides a way to sample energy values from the exponential distribution that preserves α-differential privacy. In order to sample synthetic sets (and not just energy values), we must then be able to sample uniformly from the space of sets that satisfy a particular energy distance to the data. This section defines the sensitivity of the energy distance, and proposes a method for sampling in the space of sets.

The sensitivity of the energy distance on sets in $\mathbb{R}^d$ for neighboring data sets each of size N is $\Delta f = \frac{2d^{1/p}(2N-1)}{N^2}$. By setting the score function of the exponential mechanism to be the negative energy distance between data and synthetic sets, we induce an exponential distribution over sampled energy values. Sections 5.3.2 and 5.3.3 provide more detail.

Synthetic sets are sampled using the Metropolis-Hastings algorithm, where they are perturbed one point at a time using Gaussian noise and a tailored step size, and are accepted based on the ratio of exponential term values. Section 5.3.4 provides more detail.

Throughout this work, data and synthetic set sizes are denoted as N and n, respectively. This notation accommodates the setting of this section, where $n = N$, while demonstrating that results generalize to settings of varying set sizes.

5.3.2 Sensitivity

For neighboring data sets $X = \{x_1, \ldots, x_{N-1}, x_N\}$ and $X' = \{x_1, \ldots, x_{N-1}, x'_N\}$ and candidate set $Y = \{y_1, \ldots, y_n\}$, with all points in space $\mathcal{D} \in [0, 1]^d$, the sensitivity of the energy distance is defined as

$$|e(X',Y) - e(X,Y)| = \left| \frac{2}{Nn}\sum_{j=1}^{n}\left(||x'_N - y_j||_p - ||x_N - y_j||_p\right) - \frac{1}{N^2}\left[\sum_{j=1}^{N}\left(||x'_N - x_j||_p - ||x_N - x_j||_p\right) + \sum_{i=1}^{N}\left(||x_i - x'_N||_p - ||x_i - x_N||_p\right)\right]\right|. \quad (5.7)$$

Without loss of generality, the first term is positive and maximized when $x'_N = [\mathbf{0}]_d$, $x_N = [\mathbf{1}]_d$, and $y_j = [\mathbf{1}]_d \ \forall j$. In the first term, the maximum value within parentheses is therefore $d^{1/p} - 0 = d^{1/p}$. With the same values of x_N and x'_N, the second term is minimized when $\forall i \neq N$, $x_i = [\mathbf{0}]_d$. In the second term, the minimum value within each pair of parentheses is therefore $0 - d^{1/p} = -d^{1/p}$, with one exception. The N'th terms in each summation represent norms on the diagonal of the Gram matrix, and are therefore equal to zero. To summarize, the sensitivity is calculated as

$$\Delta f = \left| \frac{2}{Nn}\sum_{j=1}^{n}\left(d^{1/p} - 0\right) - \frac{1}{N^2}\left[\sum_{j=1}^{N-1}\left(0 - d^{1/p}\right) + \sum_{i=1}^{N-1}\left(0 - d^{1/p}\right)\right]\right| \quad (5.8)$$

$$= \frac{2d^{1/p}}{N} + \frac{2d^{1/p}(N-1)}{N^2} \quad (5.9)$$

$$= \frac{2d^{1/p}(2N-1)}{N^2}. \quad (5.10)$$

Based on this form, we note that the sensitivity is dependent not on the size n of the candidate set, but only on the size N of the data set.

Remark 1 (Equal size). When the data and candidate sets have the same size, i.e. $N = n$, sensitivity equals $\frac{2d^{1/p}(2n-1)}{n^2}$. This will be used in later methods that utilize support points.

Remark 2 (MMD sensitivity). When using the maximum mean discrepancy (MMD), as opposed to the energy distance, the sensitivity follows similarly as $\frac{2K(2N-1)}{N^2}$, for bounded kernel $k(\cdot,\cdot) \in [0,K]$. For the radial basis function kernel, $k(x,y) = \exp(-\frac{1}{2\sigma}||x-y||^2)$, $K = 1$ yielding sensitivity $\frac{2(2N-1)}{N^2}$.

5.3.3 Worked Example

For ease of description, consider the case with $N = 4, n = 3$, $d = 1$, and $p = 1$:

$$X' = \{0, 0, 0, 0\}, X = \{0, 0, 0, 1\}, Y = \{1, 1, 1\} \tag{5.11}$$

$$e(X', Y) = \frac{2}{4 \cdot 3} \sum \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{\text{“}xy\text{”}} - \frac{1}{4^2} \sum \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{“}xx\text{”}} - \frac{1}{3^2} \sum \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{“}yy\text{”}} \tag{5.12}$$

$$e(X, Y) = \frac{2}{4 \cdot 3} \sum \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{4^2} \sum \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} - \frac{1}{3^2} \sum \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.13}$$

The sensitivity consists of two parts: the difference between “xy” Gram matrices (the first terms above), and the differences between “xx”-Gram matrices (the second terms above). Sensitivity is maximized when the absolute values of these differences maximized. Below, we show the case for positive differences. The largest-magnitude negative difference occurs trivially from swapping X' and X.

For the first part, since X' and X differ by only one point, the “xy” matrices differ by only one row. In this example, the last row changes, and the maximum value of each norm, i.e. $||x_N - y_j||_1$, is equal to the diameter of the domain, or one in this case. (In general, for data on $[0, 1]^d$, $\max || \cdot ||_p = d^{1/p}$.) The first term of the sensitivity is therefore maximized at $(2/N \cdot n) \cdot 1 \cdot n = (2/4 \cdot 3) \cdot 3 = 1/2$.

For the second part, consider the negative sign as part of the matrix term. The difference is maximized when norms in the X' term are minimal

(i.e. zero, thus maximizing the overall term) and norms in the X term are maximal (i.e. one, thus minimizing the overall term). What remains is to solve for the largest number of elements in the X term that can be equal to one. For sets of size N with $N-k$ zeros and k ones, the number of elements with value equal to one is $2k(N-k)$. To maximize the difference, we solve the following expression with k and $k+1$, from $k=0$ to $k=N-1$:

$$\text{(case } X') \qquad -2k(N-k) = 2k^2 - 2kN \tag{5.14}$$

$$\text{(case } X) \qquad -2(k+1)(N-(k+1)) = 2k^2 - 2kN + 4k - 2N + 2 \tag{5.15}$$

$$\text{diff}_{k,k+1} = -4k + 2N - 2 \tag{5.16}$$

$$\underset{k}{\text{argmax}}\ \text{diff} = 0. \tag{5.17}$$

The second term of the sensitivity is therefore maximized at $(1/N^2)\cdot 1\cdot(2N-2) = (1/4^2)\cdot 1\cdot(2\cdot 4-2) = 3/8$. Collecting terms, the energy sensitivity is therefore $1/2 + 3/8 = 7/8$.

5.3.4 Sampling Synthetic Sets

Let $\mathcal{X}$ be the space of sets of size N, where each set has points in $\mathcal{D}$. With a sensitivity value Δf and data X, a synthetic set $\widetilde{X}$ can be sampled using the exponential mechanism [95] with negative energy as the score function:

$$f(\widetilde{X}) = f(\widetilde{X}, X) = -e(\widetilde{X}, X) \tag{5.18}$$

$$\Pr[\widetilde{X}] = \Pr[\tilde{e}] \ \propto\ \exp\Big\{\frac{\alpha}{2}\frac{f(\widetilde{X})}{\Delta f}\Big\} = \text{Exp}\Big(\frac{2\Delta f}{\alpha}\Big). \tag{5.19}$$

Critically, the link between sampling an energy value in $\mathbb{R}$ and sampling a synthetic set in $\mathcal{X}$ must be established. This is not trivial, since exploration of the $[0,1]^{N\times d}$ sample space may be complex.

Sample using Metropolis-Hastings To sample from $\mathcal{X}$, consider establishing a uniform prior over that space and using the score function above (the negative energy distance) as a form of likelihood.

The Metropolis-Hastings algorithm would apply as follows. First sample an initial candidate synthetic set X_t, and perturb those points in a random direction to get X'_t. Let $p(\cdot)$ be the prior, and let the acceptance ratio be $\gamma = {}^{f(X'_t)}/_{f(X_t)}$, such that f is proportional to the true density. The acceptance ratio takes the form

$$\gamma = \frac{f(X'_t)p(X'_t)q(X_t|X'_t)}{f(X_t)p(X_t)q(X'_t|X_t)} = \frac{\exp\left\{\frac{\alpha}{2}\frac{f(X'_t)}{\Delta f}\right\}}{\exp\left\{\frac{\alpha}{2}\frac{f(X_t)}{\Delta f}\right\}} = \exp\left\{\frac{\alpha}{2\Delta f}\Big[f(X'_t) - f(X_t)\Big]\right\} \tag{5.20}$$

$$= \exp\left\{\frac{\alpha}{2\Delta f}\Big[e(X_t, X) - e(X'_t, X)\Big]\right\}. \tag{5.21}$$

Then, draw a uniform random variable ν on $[0, 1]$. If $\nu < \gamma$, accept the candidate and let $X_{t+1} = X'_t$; otherwise, reject the candidate and let $X_{t+1} = X_t$.

Although this sampling procedure will asymptotically reach the correct stationary distribution, when using finitely many samples, we assess its performance using well-known convergence diagnostics. A more detailed discussion on convergence appears in Section 5.5.

5.4 Practical Private Synthesis using Support Points

The method of the previous section can be impractical for large N, since the computational complexity of the energy distance scales with the sizes of the two point sets. When producing a "full size" synthetic set (same size as data), the Gram matrix has $(2N)^2$ elements. If N is large, this can become untenable. Some adaptations could alleviate the computational load. We might, for example, produce a smaller synthetic set, i.e. choose n such that $n << N$, reducing the number of elements to $(N + n)^2$. This adaptation brings some savings, but is still $\mathcal{O}(N^2)$.

Substantially more savings are possible if the data can first be represented by a smaller set of size n. In this setting, the size-n representation acts as the data, and the Gram matrix contains only n^2 elements. The resulting procedure thus has complexity $\mathcal{O}(n^2)$, for integer-valued $n \geq 1$. This

strategy—using support points as smaller representative points—forms the main application of this work.

5.4.1 Support Point Method and Sensitivity

In order to use the "small representation" strategy of the previous section, we must compute the sensitivity of the energy distance when support points are used in place of the typical reference data set. Fortunately, the sensitivity, logic, and entire procedure of Section 5.3 remain the same, due to the observation that a maximal shift of a single data point can produce exactly the same maximal shift in a single support point.

Maximal Data Point Shift Can Lead To Maximal Support Point Shift The sensitivity calculation of Section 5.3.2 is based on the maximal shift of a single data point across a bounded space $\mathcal{D} := [0, 1]^d$. In order to use the same logic here, we must show that the shift of a single data point can lead to the equivalent shift of a single support point. By computing the gradient of the energy distance for a single support point in a simple setting, we can infer that in certain cases, support points will follow outlying data points. Consider $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_n\}$ where $n < N$. Using the definition of energy distance in Eq. 5.5 with $d = 1$ and $p = 1$, the gradient for a single support point is

$$\frac{\partial e}{\partial y^*} = -\frac{2}{Nn}\sum_{i=1}^{N}\frac{x_i - y^*}{||x_i - y^*||} - \frac{2}{n^2}\sum_{j=1}^{n}\frac{y^* - y_j}{||y^* - y_j||}, \tag{5.22}$$

where the second term carries a factor of 2, due to the presence of y^* in both one row and one column. When one outlying point exists in X, a point in Y will find equilibrium between the outlier and the remaining points in X. Using the notation of order statistics: For data X and support points Y, if $x^{(N)} >> x^{(N-1)}$, then $y^{(n)} \in (x^{(N-1)}, x^{(N)})$. As such, the gradient of the largest

support point is

$$\frac{\partial e}{\partial y^{(n)}} = -\frac{2}{Nn}\sum_{i=1}^{N}\frac{x_i - y^{(n)}}{||x_i - y^{(n)}||} - \frac{2}{n^2}\sum_{j=1}^{n}\frac{y^{(n)} - y_j}{||y^{(n)} - y_j||} \tag{5.23}$$

$$= -\frac{2}{n}\left[\underbrace{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{1}_{x_i > y^{(n)}} - \mathbb{1}_{x_i < y^{(n)}}\right)}_{A} + \underbrace{\frac{1}{n}\sum_{j=1}^{n}\left(\mathbb{1}_{y^{(n)} > y^{j}} - \mathbb{1}_{y^{(n)} < y^{j}}\right)}_{B}\right]. \tag{5.24}$$

Based on the order statistic, $B = n - 1$, since $y^{(n)}$ is larger than all but one point in Y, i.e. $y^{(n)}$ itself. For notational ease, multiply by $\frac{Nn}{2}$, which preserves gradient directions. This yields the approximate, direction-preserving gradient

$$\square = \widetilde{\frac{\partial e}{\partial y^{(n)}}} \tag{5.25}$$

$$= -A - \frac{N}{n}B \tag{5.26}$$

$$= -A - \frac{N}{n}(n-1). \tag{5.27}$$

If $k = |\{x_i : x_i > y^{(n)}\}|$ represents the number of points in X larger than the largest support point, then $A = k - (n - k)$, and

$$\square = -k + (n-k) - \frac{N}{n}(n-1) \tag{5.28}$$

$$= -2k + \frac{N}{n} \tag{5.29}$$

For the most extreme case (all points at one corner except one outlier at the opposite corner), k equals 1 and we are interested in when $\square$ is positive versus negative. The result of Eq. 5.29 indicates that $y^{(n)}$ decreases when $\frac{N}{n} > 2$, and increases when $\frac{N}{n} < 2$. In other words, there exist cases where a support point would "chase" an outlying data point. The ability of the single support point to find an equilibrium position across the entire domain therefore establishes the sensitivity as identical to that of the data. Our sensitivity calculation is understood as optimal for $p = 1$, but loose for $p > 1$, where being far from remaining points is more heavily penalized. Solving this equilibrium position remains a question for further research.

5.4.2 Algorithm

Below we detail the full procedure for practical private synthesis.

Algorithm 1: Practical Private Synthesis

Input: Data X of size N and dimension d, privacy budget α, desired synthetic set size n, energy norm p.

Result: Synthetic representative points satisfying α-differential privacy.

Get support points as $Y^* = \arg\min_{y_1,\ldots,y_n} \; e_{N,n}(X, Y)$.
Get energy sensitivity as $\Delta f = \frac{2d^{1/p}(2N-1)}{N^2}$.
Randomly initialize candidate point set Y_t of size n.
while *not converged* **do**
    Get energy between candidate and optimal, $e_t = e(Y_t, Y^*)$.
    Perturb a single point in Y_t, giving Y_{t+1}.
    Get energy between new candidate and optimal, $e_{t+1} = e(Y_{t+1}, Y^*)$.
    Compute acceptance ratio, $r = \frac{\exp(-\alpha/2\Delta f * e_{t+1})}{\exp(-\alpha/2\Delta f * e_t)}$.
    Sample $\nu \sim U(0, 1)$.
    if $\nu < r$ **then**
        Accept candidate, $Y_t = Y_{t+1}$.
    else
        Reject candidate, continue.
    end
end
$\widetilde{Y} = Y_t$ is a set of n synthetic representative points satisfying α-differential privacy.
Optional: Expand $\widetilde{Y}$ using kernel density estimation or repeated sampling (see Section 5.6 for details).

5.5 Convergence

Although the Metropolis-Hastings sampling procedure of Section 5.3.4 converges to the stationary distribution asymptotically, we can assess finite runs of the procedure using typical convergence diagnostics. Using multiple

randomly-initialized chains, we compare within-chain and between-chain variance, and verify that a recent variant of the Gelman-Rubin diagnostic [131] approaches one.

Another sampling approach called the grid-walk algorithm [12, 5] shows how to sample proportional to a log-concave function $F(x)$, and provides bounds on the sampling error, which could inform on the β of an (α, β)-approximate differential privacy ("aDP") procedure. For convergence guarantees to apply, several requirements must be satisfied: (1) the parameter space must be convex, (2) the convex set must be in isotropic position, (3) $\ln F(x)$ must be Lipschitz, (4) $\ln F(x)$ must be convex, and (5) the sampling distribution $F(x)$ must be log-concave. These requirements can be satisfied in the following ways, respectively: (1) our parameter space is the space of support point sets, which is convex by definition, since they lie on a bounded hypercube, (2) our space of support points can be transformed to be in isotropic position, (3) on a bounded space, the Lipschitz constant for $\ln F(x)$ is equivalent to a modified version of the energy sensitivity, (4) we may use the convex relaxation of the energy distance provided in Eq. 22 of [91], and (5) the exponential construction with the negative [convex] energy is clearly log-concave. In the implementation, the grid steps of the algorithm are fixed-length increments/decrements to a single element of a single support point.

This approach, however, requires a long run-time that is considered unreasonable for all but the simplest settings. To illustrate this, consider the following setup, using only ten support points in two dimensions.

For N support points on $[0,1]^d$ and privacy budget α, let $d^* = Nd$ be the dimension when point coordinates are vectorized. For neighboring sets X and X', and support points Y: $X, X', Y \in [0,1]^{N\times d}$, let Δf be the sensitivity of the energy distance as defined in Eq. 5.10.

We aim to sample proportional to $F(X)$ where

$$F(X) = \exp\Big(-\frac{\alpha}{2}\frac{e(X,Y)}{\Delta f}\Big), \tag{5.30}$$

$$f(X) = \ln F(X) = -\frac{\alpha}{2\Delta f}e(X,Y). \tag{5.31}$$

The following Lipschitz condition must hold:

$$|f(X) - f(X')| \le L\Big(\max_{i=1}^{n}|X_i - X_i'|\Big). \tag{5.32}$$

This is reminiscent of the energy sensitivity, but rather than changing one point maximally, only one coordinate changes. A modified sensitivity is defined as

$$\Delta f^* = \left| \frac{2}{N^2}\sum_{j=1}^{N}\left(1^{1/p} - 0\right) - \frac{1}{N^2}\left[\sum_{j=1}^{N-1}\left(0 - 1^{1/p}\right) + \sum_{i=1}^{N-1}\left(0 - 1^{1/p}\right)\right]\right| \tag{5.33}$$

$$= \frac{2(2N-1)}{N^2}. \tag{5.34}$$

With $\max_{i=1}^{n}|X_i - X_i'| = 1$, the maximal change is

$$|f(X) - f(X')| \le \left| -\frac{\alpha}{2}\cdot\frac{\Delta f^*}{\Delta f}\right| = \frac{\alpha}{2d^{1/p}} = L. \tag{5.35}$$

A few more definitions are required: The cube size is defined as $\delta = 1/\lceil L\rceil$. Let γ be the slack parameter of concavity, assumed to be zero for concave relaxation of negative energy. Let β be the second parameter of (α,β)-aDP. Let the probability of starting at a random position on the grid be $q(X_0) = 1/\lceil L\rceil^{d^*}$. All together, we can compute the number of steps, defined in Section 4 of [5]:

$$16\exp(4)d^{*2}X_{\max}^2L^2\exp(2\gamma)\Big(\log\frac{\exp(2)}{\beta} + \log\frac{1}{q(X_0)}\Big). \tag{5.36}$$

For $N = 10$, $d = 2$, $X_{\max} = 1$, $\alpha = 100$, $p = 2$, $\gamma = 0$, and $\beta = 1/N$, the number is $\sim$33B.

In addition to the grid-walk algorithm, [87] and the references therein also address the task of sampling from densities proportional to (near) log-concave functions, and provide alternatives. Exploration and comparison of sampling methods in the setting of private support points remain open avenues for future research.

5.6 Expansion of Support Points to Full Synthetic Data Set

Support points can be expanded into larger synthetic data sets using kernel density estimation or using repeated sampling.

5.6.1 Kernel Density Estimation

The simplest form of expansion is kernel density estimation, where a Gaussian mixture model is made with kernels centered at each support point. For private support points $\widetilde{Y} = \{\tilde{y}_1, \ldots, \tilde{y}_n\}$, we draw a full size synthetic data set $\widetilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_N\}$ from the following distribution:

$$\tilde{x} \sim \text{GMM}(x \,|\, \widetilde{Y}, \sigma) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{N}(x \,|\, \tilde{y}_i, \sigma), \tag{5.37}$$

where the variance (i.e. bandwidth) of each kernel is selected in some data-independent, or differentially private manner. Several options exist.

A data-independent bandwidth parameter can be selected using prior knowledge from experts or estimated using an independent data set.

Alternatively, the maximum likelihood estimate (MLE) of the bandwidth can be computed by evaluating over a range of bandwidths, and computing the data likelihood for each. By setting bounds on the bandwidth, a differentially private version of the MLE bandwidth can be sampled using the Laplace mechanism and used in the expansion step. To ensure a positive bandwidth, the resulting value can be clipped[1] to be above a pre-set minimum.

[1] See [39, 31, 22, 11] for references on post-processing and bounding.

A third option is to use the setup just described, but rather than perturb the MLE bandwidth, instead select from the finite set of bandwidths proportional to the likelihood — i.e. use the exponential mechanism with likelihood as the score function.

Note that the second and third methods access the data again to compute likelihoods, and therefore add to the overall privacy budget. Privacy budget growth for sequential data access is addressed by composition theorems of differential privacy. See [96, 67] for more details.

5.6.2 Repeated Sampling

In settings where bandwidths cannot be estimated in a data-independent manner or via a differentially private mechanism, a synthetic data set can be generated by repeated sampling of private support points. By the composition theorem of differential privacy [96, 67], k repeated samples from an α-differentially private sampling mechanism yield a $k\alpha$-differentially private sample.

One advantage of repeated samples is the ability to produce uncertainty estimates by computing a desired function over several independently privatized samples. The model-based synthesis of [83] uses this principle in a Bayesian model setting, by repeatedly sampling private sufficient statistics, then sampling a synthetic set from each [now-private] posterior predictive distributions. As before, such methods face the risk of inflating the privacy budget, or risk distributing a fixed budget among many synthesis steps, such that each step contains less information.

5.7 Experimental Results

5.7.1 Gaussian Mixture Model

The following section provides empirical examples of each of the above methods using data from weighted Gaussian mixture models in two dimensions. As the number of support points and the privacy budget decrease, we expect that these methods will produce less informative samples. In contrast,

as the number of support points and the privacy budget increase, we expect that samples will accurately reflect the data.

5.7.1.1 Data

We adapt the data-generating code of [10] to produce bounded data on $[0,1]^d$. In this setting, for positive bandwidth σ, Gaussian data $\pm 4\sigma$ is bounded on $[0,1]^d$ by restricting cluster centers to $[0+4\sigma, 1-4\sigma]^d$, and choosing bandwidth in $(0, 1/8]$.

5.7.1.2 Support Points Optimization

Support points effectively capture the structure of complex patterns in data sets with a variety of cluster counts. Figure 5.1 demonstrates that even for highly irregular distributions, support points capture all meaningful modes, and have the "space-filling" [91] property of spreading out over areas of more uniform density. Support points are therefore considered a viable target for *private* support points, as they sufficiently and succinctly capture the data distribution.

For optimization, we use TensorFlow's implementation of the Adam optimizer, with learning rate 0.01. With non-specialized hardware such as an Intel Core i5 processor, for 200 data points and 10 support points in two dimensions, convergence is typically reached within 300 iterations and two seconds.

5.7.1.3 Sample using Metropolis-Hastings

The above sampling technique produces private support points that can be useful for estimation and consistent with the distribution of energy distances defined by the exponential mechanism. Figure 5.2 demonstrates the trade-off between privacy and accuracy, where private support points are shown to converge to true support points as the privacy budget increases. The distribution of sampled energies and the sampler performance are shown in Fig. 5.3.

For this method, we implement a simple Gaussian random walk Metropo-

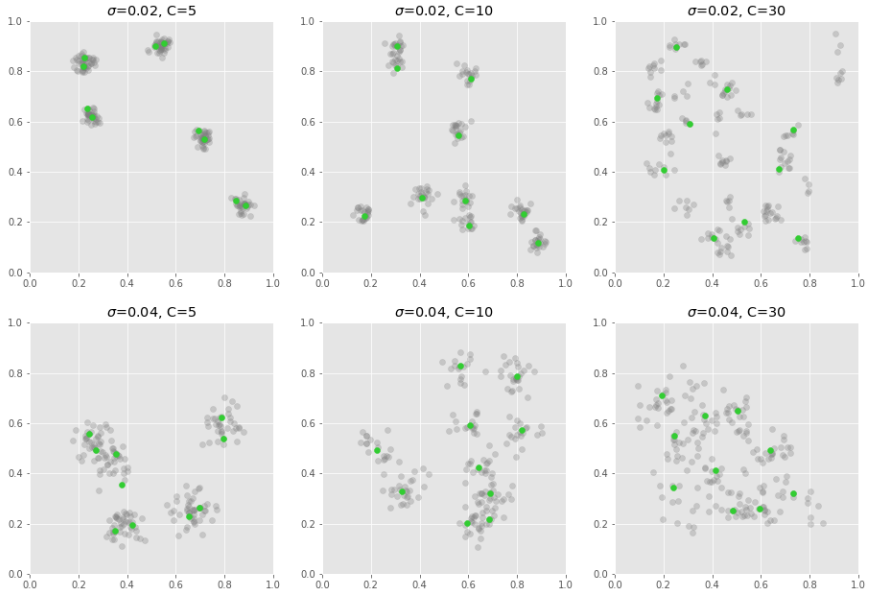


Figure 5.1: Support points (green) accurately capture complex clustering structures in data (gray) over a range of cluster variance and cluster count. In each setting, 200 data points are represented by 10 support points.

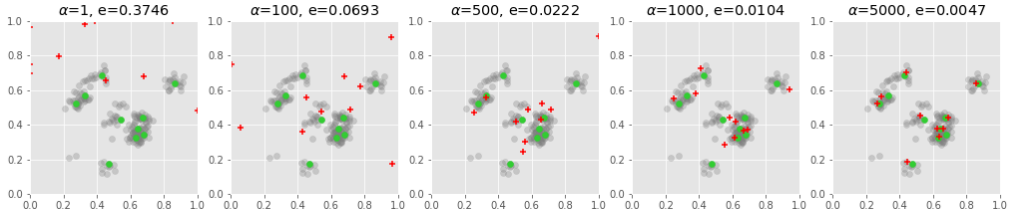
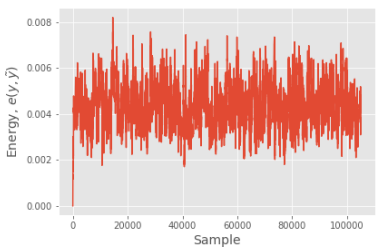
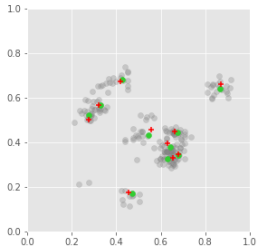


Figure 5.2: As the privacy budget α increases, privately sampled support points (red) more accurately represent data (gray) by moving closer to true support points (green). In each setting, 200 data points are represented by 10 private support points.

lis Hastings, using the acceptance criteria of Eqs. 5.20 and 5.21. We use a similar setup as in previous examples with the following configuration: 200 data points, 10 support points, adaptive step size tuned to yield a 30% acceptance



(a) Traceplot of MH energies.



(b) Data (gray), support points (green), and private samples (red), plotted for comparison.

Figure 5.3: As a baseline, the Metropolis-Hastings procedure is tested under a minimally private setting with $\alpha = 5000$. The sampling scheme mixes well, and captures the distribution.

rate, burn 5000 iterations, and thin every 2000 iterations. To establish a baseline performance, we set $\alpha = 5000$, and show a typical sample in Fig. 5.3b. Through repeated experiments, we found that when tightening the privacy budget, Metropolis Hastings samples quickly diverged to non-informative samples and did not return to higher probability regions of the posterior.

5.7.1.4 KDE with Pre-Selected Bandwidth

With privately sampled support points, a full data set can be sampled from the kernel density estimate described in Eq. 5.37. Figure 5.4 shows the result of this sampling procedure for various pre-selected bandwidths.

5.7.1.5 KDE with DP-MLE Bandwidth

Based on the data bounds, the data bandwidth must lie in $(0, 1/8]$, implying the sensitivity $\Delta f = 1/8$. We sample a DP bandwidth $\tilde{\sigma} \sim \text{Lap}(\sigma, \frac{1/8}{\alpha})$, and enforce a non-trivially small positive floor value δ, with $\max(\tilde{\sigma}, \delta)$. Here $\delta = 0.001$.

In experimental settings where noisy support points remain close to clusters, the MLE bandwidth can be orders of magnitude lower than the data-generating bandwidth. This occurs when one or more data points align with

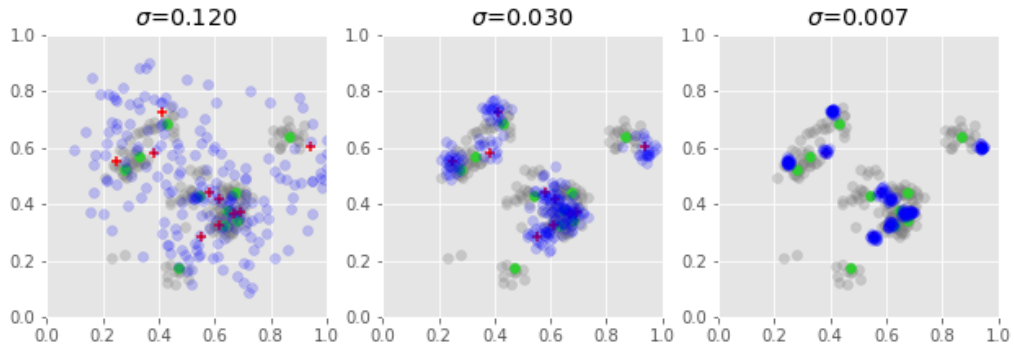
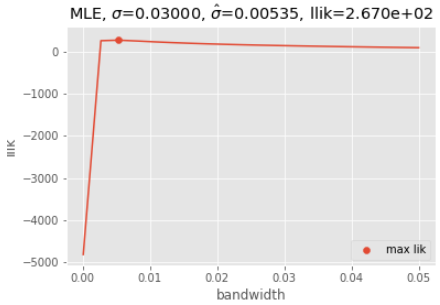


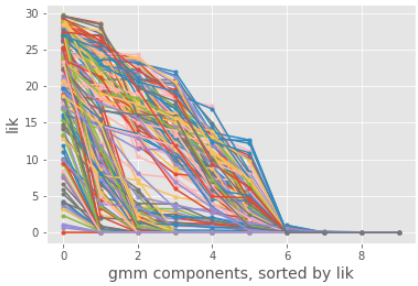
Figure 5.4: Full samples (blue) from a kernel density estimator centered on privately sampled support points (red), using various pre-selected bandwidths. The result approximates the data distribution (gray). Results are shown for $4\sigma, \sigma$, and $\sigma/4$.

very peaked (low-bandwidth) kernels, contributing a large amount to the likelihood. In contrast, for privately sampled support points that lie far from all data, the MLE bandwidth tends to be larger than the data-generating bandwidth, causing the expanded sample to be disperse and non-informative.

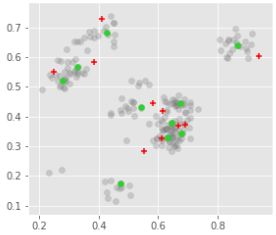
Figure 5.5 shows an intermediate setting where privately sampled support points only partially coincide with the data distribution. Figure 5.5a shows the log likelihoods of data for a range of bandwidths. At sufficiently low bandwidths, certain data point likelihoods are small enough to dominate the very negative total likelihood. Above a certain level, wider kernels cover more data points, but the assigned likelihood values are more moderate. In Fig 5.5b, each line represents one data point, where values are the likelihood per mixture model component. In the case of a data point near a very peaked kernel, this plot exhibits a high likelihood for that component, and a steep dropoff thereafter. Figure 5.5c and Fig. 5.5d show a privately sampled support point set and the full expansion using the MLE bandwidth, respectively. Figure 5.5e shows four distinct samples of full synthetic data sets, each using a differentially private bandwidth value.



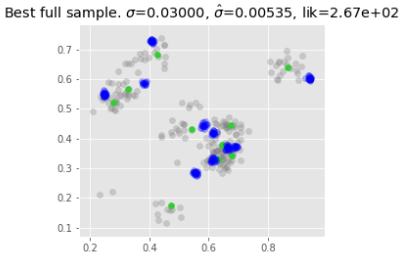
(a) The non-private MLE bandwidth can be found by evaluating data likelihood over a range of bandwidths.



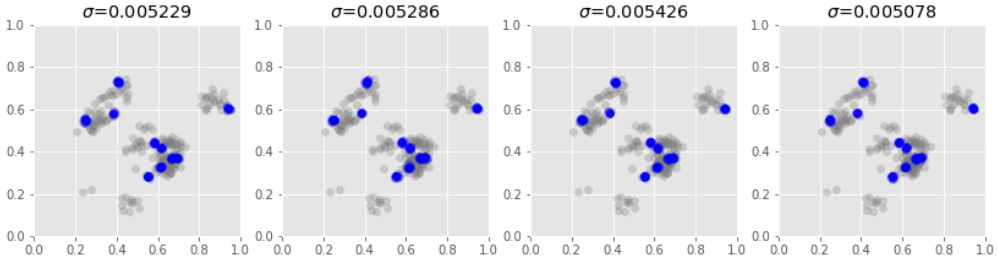
(b) Each line represents a data point, where values are likelihoods per mixture component, sorted in decreasing order.



(c) Privately sampled support points.



(d) Expansion with MLE bandwidth.



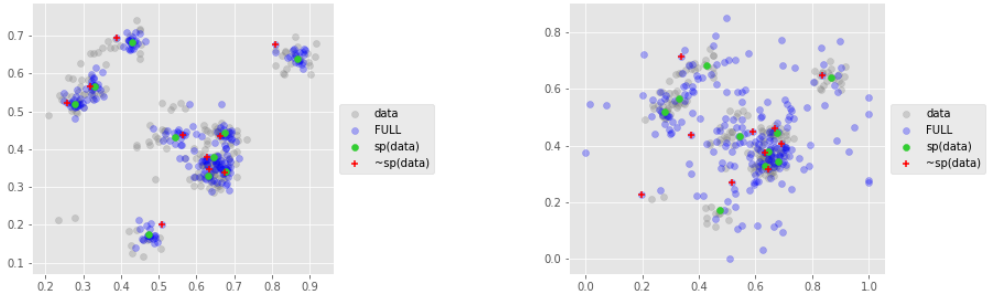
(e) Samples of full synthetic data sets from kernel density estimators with differentially private bandwidths.

Figure 5.5: A private support point set can be expanded using a kernel density estimator, using a private version of the maximum likelihood bandwidth.

5.7.1.6 Repeated Sampling

For the running example, we plot the results of repeatedly sampling private support points in Fig. 5.6. Since the total privacy budget is $k\alpha$ for k

samples each with budget α, this might be useful in settings with low sensitivity, where low-budget queries are still minimally informative.



(a) High budget yield more representative samples.

(b) Low budget yields more disperse samples.

Figure 5.6: Repeated samples of support points can be collected to yield a full sized synthetic data set. Since the privacy budget accumulates with each repetition, repeated samples might be useful for settings with low sensitivity, where individual samples maintain utility.

5.7.2 Regression Performance

Whereas previous examples focused on distribution matching, the following examples evaluate accuracy in a regression setting, and compare against existing methods. Examples here are more varied and higher dimensional than in the previous section, and contain a mixture of real-valued and ordinal data.

To our knowledge, the closest related method is Algorithm 2 of [10] (hereafter "KME"), which produces a weighted set of synthetic points whose random-feature kernel mean embedding is close to a privatized embedding of the data. In our comparisons, we include the original KME method with and without noise, and since our method is unweighted, we also include an adapted uniformly-weighted version of the KME method.

The full list of comparisons is as follows: full training data, Laplace-perturbed histogram counts (using the extended uniform grid and total bin count proposed in [122]), support points, private support points, non-private weighted KME, private weighted KME, and private uniform KME. In each

case, a portion of data is held out for evaluation, and the remaining data is used to generate a synthetic set. That synthetic set is used to fit a simple linear regression, and mean squared error (MSE) on held out data is recorded. For all methods based on histograms, support points, and KME, we test privacy budgets of 100, 1000, and 10000. For methods based on support points and KME, we also test a range of point set sizes, i.e. 5%, 10%, and 20% of training data size. In all settings, the median and interquartile range of MSEs on heldout data are reported over all results after five-fold cross-validation. Results are shown in Fig. 5.8.

In most cases, MSE scales negatively with both privacy budget and set size. Intuitively, a higher privacy budget is expected to better preserve the data distribution, making it more useful for regression (lower error); and larger point sets are expected to be more informative, also yielding lower error. Most models show a larger decline in performance when the set size decreases from 10% to 5%, compared to from 20% to 10%. This suggests that these data distributions might be most efficiently summarized (for the purposes of regression) using about 10% of the data.

Across all three data sets, perturbed histograms perform well when their size fits in memory – otherwise, results are omitted. Using the bin count framework from [122], the number of bins is $(\frac{N\alpha}{10})^{\frac{2d}{2+d}}$, for N data points of dimension d and privacy budget α. The number of bins scales positively with privacy budget, data size, and dimensionality. Larger sets, of higher dimension, and with higher budgets, can therefore lead to memory issues when storing histogram counts.

With three data sets, three representative set sizes, and three budgets, a total of 27 settings are evaluated. Private support points perform better, on par with, and worse than competing models in 20, 7, and 0 settings, respectively. The on par cases tended to be those with lower budgets, indicating that methods tended to degraded somewhat equally with large amounts of noise. Private support points also appear to perform better as the dimension of data increases. The relative improvement in MSE grows with dimensionality across all methods, sizes, and budgets.

Taken together, private support points perform well with moderate privacy budgets and comparatively better than all tested alternatives for higher

dimensional data. This might be due to the relative efficiency of support points in communicating distribution information, as evidenced by the high performance of non-private support points.

To assess convergence in this setting, we plot multiple chains initialized uniformly at random over the support of the data, and plot between- and within-chain variance, as well as a recent variant of the Gelman-Rubin convergence diagnostic [131]. Figure 5.7 illustrates these results for a run on the California data set with support set size 35 and budget 1000.

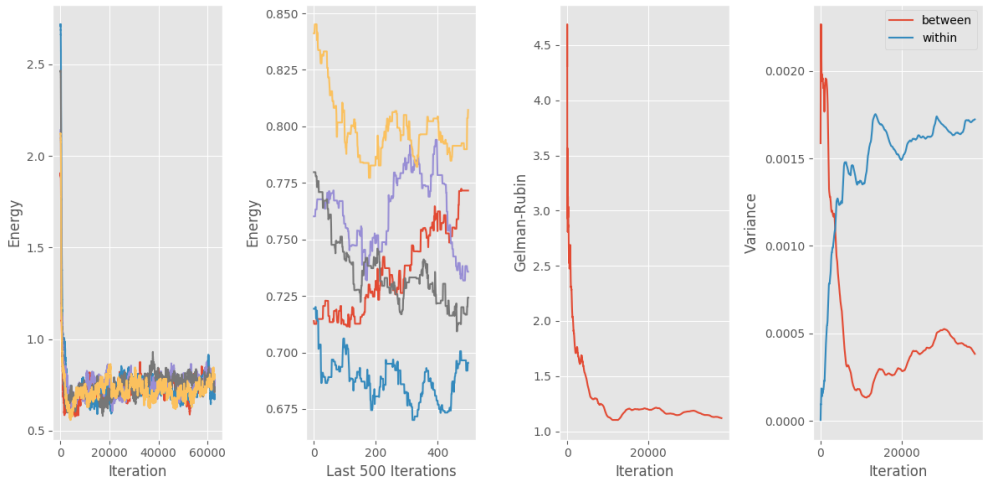


Figure 5.7: Metropolis-Hastings convergence diagnostics show multiple, randomly initialized chains converging around a mean energy distance. The Gelman-Rubin statistic trends toward one, and between-chain variance trends toward zero. The example illustrated is for support point size 40 and privacy budget 1000.

5.8 Related work

Data release can be interactive and relatively narrow – via repeated summary-style queries over a database – or can be non-interactive and full, via a one-time anonymized disclosure. Non-interactive release methods include k-anonymity [126], l-diversity [90], and t-closeness [80], which measure

and aim to regulate the distribution of sensitive attributes in a data set. Other methods include several variations of the Laplace, Exponential, and Gaussian mechanisms applied to histograms. See [73] and [22] for a review. This work, along with several existing methods, utilizes the dimensionality reduction approach of applying differential privacy to a usually-simpler representation of the entire data set.

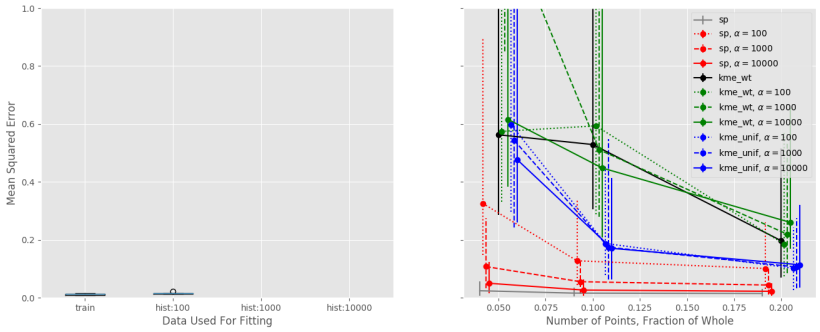
In [79] and [9], a copula model is formulated, computing differentially private versions of marginal histograms and a differentially private correlation matrix. [138] utilizes random projections of high-dimensional data, where performance depends on projection dimensionality. [82] adopts a compressed sensing approach where Laplace noise is added to compressive samples before reconstruction. Similarly, for contingency tables, [137] propose to add Laplace noise to coefficients of data transformed using the Haar wavelet transform. Another approach in [42] builds private coresets, using differentially private medians on projected lines in a Voronoi partition based on a coreset-like point set called "bi-criteria" centers. In terms of mechanism setup, our use of the negative energy distance as a score function in the exponential mechanism is close in spirit to the negative error score function of [20].

Finally, as mentioned in Sec. 5.7.2, our use of the energy distance, a form of Maximum Mean Discrepancy [52], relates to the synthetic database work of [10], where weighted synthetic points minimize distance in the space of empirical kernel mean embeddings, computed either by projection onto an orthonormal basis or onto random Fourier features. In that work, synthetic points are weighted, allowing negatively-weighted points to be positioned far from the data distribution, indicating where *not* to be. Synthetic points may therefore provide utility for function estimation, but be less useful as credible synthetic points for other applications.

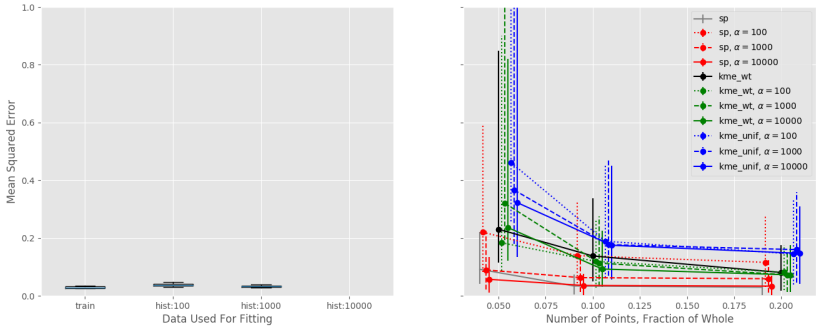
5.9 Conclusion

We have demonstrated a novel method of differentially private data release using support points. By applying the exponential mechanism with the negative energy distance as the score function, we can produce samples of support points which produce the desired exponential distribution of scores.

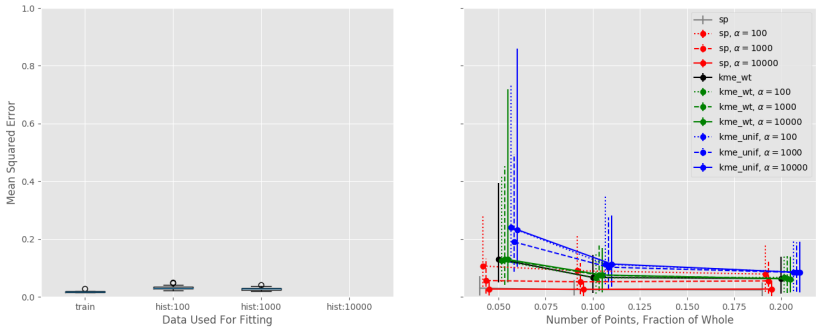
We detail the existence of a theoretically valid Metropolis Hastings sampling procedure, and evaluate the sampling procedure on varied examples. Experiments illustrate the anticipated behaviors, where larger support point sets and larger privacy budgets produce more accurate samples, and vice versa. Scaling such procedures and exploring improved sampling schemes remain open questions for future research.



(a) Boston, n=506, d=14



(b) Diabetes, n=442, d=11



(c) California, n=500*, d=9

Figure 5.8: Private support points outperform similar methods in all but the lowest-budget setting (where performance is on par), and comparative advantage grows with dimensionality. Error bars and central dots represent interquartile ranges and medians, respectively. *A random subset of 500 points is chosen for training in the California dataset.

Chapter 6

Conclusion

Characterizing distributions from finite data is core to the field of statistics, and is the basis for many empirical scientific pursuits. Importantly, data collection and modeling ideally yield general (and not sample-specific) insights. In this way, the notions of a generative modeling and privacy appear complementary — both seek to communicate general population-level information without repeating individual training samples.

This work expands this discussion by (1) demonstrating a new way to flex a generative model's output distribution away from the sample distribution, (2) providing a practical evaluation of generative model privacy in a medical data setting, and (3) introducing a new way of applying differential privacy guarantees to a useful data reduction tool called support points.

Future lines of research might include extensions of importance weights to other distribution metrics for modifying generative model output, and further characterization of support point sensitivity for cases with higher order norms.

Appendices

Appendix A

Importance Weighted Generative Networks

A.1 Proof of Theorem 1

Before we prove Theorem 1, we will define some notation. Suppose $p = \{p_1, ..., p_m\}$, $x = \{x_1, ..., x_m\}$ and $y = \{y_1, ..., y_m\}$ are the empirical samples obtained from $\mathbb{P}$, $M\mathbb{P}$ and $\mathbb{Q}$, respectively. We use the following quantity as in [51], with samples p and y:

$$h(z_i, z_j) = k(p_i, p_j) + k(y_i, y_j) - k(p_i, y_j) - k(p_j, y_i). \tag{A.1}$$

Here, $z_i = (p_i, y_i)$ denotes a pair of i.i.d. samples from $\mathbb{P} \times \mathbb{Q}$. The estimator $\widehat{\mathrm{MMD}}^2(\mathbb{P}, \mathbb{Q})$ can be written as

$$\widehat{\mathrm{MMD}}^2(\mathbb{P}, \mathbb{Q}) = \frac{1}{m(m-1)} \sum_{i \neq j} h(z_i, z_j).$$

Proof. Now consider the setting with samples x and y. For a modifying function $M(\cdot)$ with values on $(0, 1]$, the weights $w(x_i) = 1/M(x_i)$ are therefore bounded above, *i.e.* $1 \leq w(x_i) \leq W$. We rewrite the function h, now including weights, as

$$h'(z_i, z_j) := w(x_i)w(x_j)k(x_i, x_j) + k(y_i, y_j) - w(x_i)k(x_i, y_j) - w(x_j)k(x_j, y_i). \tag{A.2}$$

Assuming the kernel $k(\cdot, \cdot)$ is bounded between 0 and K, we can infer function bounds such that $-2WK \leq h'(z_i, z_j) \leq K(W^2 + 1)$.

Using Theorem 10 from Gretton et al. [51], we have that

$$P(\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P},\mathbb{Q}) - \mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) > t) \leq \exp\left(\frac{-2t^2 m_2}{((K(W^2+1)-(-2WK))^2}\right) = \exp\left(\frac{-2t^2 m_2}{K^2(W+1)^4}\right), \tag{A.3}$$

where $m_2 := \lfloor \frac{m}{2} \rfloor$, as the MMD requires two samples to evaluate $h(z_i, z_j)$. □

A.2 Proof of Theorem 2

Before we prove Theorem 2, we prove two functional lemmas.

Lemma 1. The variance of the estimator $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P},\mathbb{Q})$ given m samples each from $M\mathbb{P}$ and $\mathbb{P}$ is upper bounded by $2\sigma^2/m$, where $\sigma^2 = \mathrm{Var}(h(Z_i, Z_j))$ and $Z_i \sim M\mathbb{P} \times \mathbb{Q}$.

Proof. Let $\sigma^2 = \mathrm{Var}(h(Z_i, Z_j))$ and let $\sigma_1^2 = \mathrm{Var}(\mathbb{E}[h(Z_i, Z_j)|Z_i = z_i])$. Using Hoeffding's Theorem and the fact that $2\sigma_1^2 \leq \sigma^2$ [58], we bound the variance of the unbiased MMD U-statistic by

$$\begin{aligned}
\mathrm{Var}(\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P},\mathbb{Q})) &= \frac{1}{\binom{m}{2}} \sum_{c=1}^{2} \binom{2}{c}\binom{m-2}{2-c}\sigma_c^2 \\
&\leq \frac{1}{\binom{m}{2}} \left[2(m-2)\sigma_1^2 + \sigma^2\right] \\
&\leq \frac{2}{m(m-1)} \left[(m-1)\sigma^2\right] = \frac{2\sigma^2}{m}.
\end{aligned}$$

□

Lemma 2. We have the following bound:

$$\mathrm{Var}(h(Z_i, Z_j)) \leq 5\left(K^2\left(\mathbb{E}\left[\frac{1}{M(X)^2}\right] + 1\right)^2 + \mathrm{MMD}^4(\mathbb{P},\mathbb{Q})\right),$$

where the expectation is with respect to the distribution $M\mathbb{P}$.

Proof. Let $\mu = \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q})$. Note that $\mathbb{E}[h(Z_i, Z_j)] = \mu$. Therefore, we have the following chain,

$$
\begin{aligned}
&\mathrm{Var}(h(Z_i, Z_j)) \\
&= \mathbb{E}[(h(Z_i, Z_j) - \mu)^2] \\
&= \mathbb{E}\left[\left(\frac{k(X_i, X_j)}{M(X_i)M(X_j)} + k(Y_i, Y_j) - \frac{k(X_i, Y_j)}{M(X_i)} - \frac{k(X_j, Y_i)}{M(X_j)} - \mu\right)^2\right] \\
&= 25\mathbb{E}\left[\left(\frac{k(X_i, X_j)}{5M(X_i)M(X_j)} + k(Y_i, Y_j)/5 - \frac{k(X_i, Y_j)}{5M(X_i)} - \frac{k(X_j, Y_i)}{5M(X_j)} - \frac{\mu}{5}\right)^2\right] \\
&\leq 25\mathbb{E}\left[\frac{1}{5}\left(\frac{k(X_i, X_j)^2}{M(X_i)^2M(X_j)^2} + k(Y_i, Y_j)^2 + \frac{k(X_i, Y_j)^2}{M(X_i)^2} + \frac{k(X_j, Y_i)^2}{M(X_j)^2} + \mu^2\right)\right] \\
&\leq 5\mathbb{E}\left[\frac{K^2}{M(X_i)^2M(X_j)^2}\right] + 5K^2 + 10\mathbb{E}\left[\frac{K^2}{M(X_i)^2}\right] + 5\mu^2
\end{aligned}
$$

This implies the lemma as X_i, X_j are independent and generated from $M\mathbb{P}$. The first inequality follows from the fact that $(\sum_i p_i a_i)^2 \leq \sum_i p_i a_i^2$, if p lies on the simplex. The last inequality follows from the assumption that $|k(.,.)| \leq K$. □

Proof of Theorem 2. Define $\tilde{\sigma}^2$ to be the variance upper bound in Lemma 2. Suppose we have m samples from $M\mathbb{P}$ and $\mathbb{Q}$, $z_i = (x_i, y_i)$ for $i = 1, ..., m$. We divide the samples into $k = 8\log(1/\delta)$ groups, where $\log(1/\delta) = mt^2/64K^2\sigma^2$. We form the estimators of type $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})$ for each of the groups indexed $l = 1, ..., k$. Let $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(l)}$ be the estimator for group l.

Note that by Lemma 1 the variance of $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(l)}$ is bounded by $2k\tilde{\sigma}^2/m$. Therefore, with probability at least $3/4$, $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})^{(l)}$ is within $2\times\sqrt{2k\tilde{\sigma}^2/m}$ distance of its mean. As such, the probability that the median is not within the distance $2 \times \sqrt{2k\tilde{\sigma}^2/m}$ is at most $\mathbb{P}(\mathrm{Bin}(k, 1/4) > k/2)$, which is exponentially small in k. Substituting the value of k yields the result. □

A.3 Implementation and Additional Experiments

A.3.1 Synthetic Data

For the synthetic data experiment of Section 3.3.1, we show the full results in Table A.1 and Table A.2 for three discrepancy measures: squared MMD, energy distance, and estimated KL divergence. We note that the squared MMD used in evaluation is the standard estimator.

A.3.2 Yearbook

The C-DCGAN is trained for 25 epochs using the ADAM optimizer with $\alpha = 2\,\mathrm{e}{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$, and a batch size of 64. The latent variable has dimension 100, and we condition on a 22-dimensional vector corresponding to each half-decade in the dataset.

Networks for the importance weighted and median of means estimator are trained using and RMSprop optimizer with learning rate $5\,\mathrm{e}{-5}$. We use the same regularizers and schedule of generator-discriminator updates as [76]. For $\widehat{\mathrm{MMD}}^2_{IW}(\mathbb{P}, \mathbb{Q})$ a batch size of 64 was used, and for $\widehat{\mathrm{MMD}}^2_{MIW}(\mathbb{P}, \mathbb{Q})$, a large batch of 128 was split randomly into 8 groups of 16 samples.

Figure A.1 shows interpolation in the latent z for the half-decade experiment in Section 3.3.2. Figure A.2 shows another Yearbook experiment with larger imbalance between 2 time periods: Old (1930) and New (1980-2013). MMD-GANs are trained for 15,500 generator iterations.

Figure A.3 shows a related experiment in which we produce more older images given a dataset with equal amounts of old (1925-1944) and new (2000-2013) photos. Here, each time period contains over 4,500 images, which increases the stability of conditional GAN training. MMD-GANs are trained until convergence (8,000 generator iterations).

A.3.3 MNIST

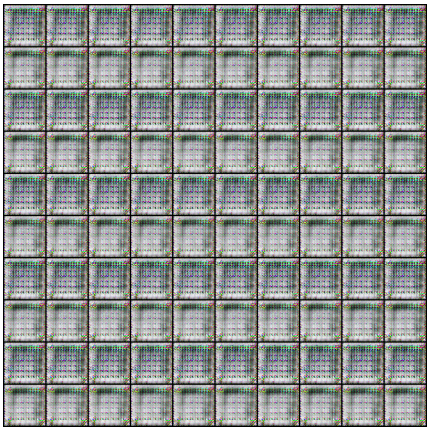
Analogous to the class rebalancing problem of Section 3.3.1, Figure A.4 shows good performance going from a balanced distribution to specific boosted levels.

Table A.1: Squared MMD, energy distance, and estimated KL divergence between generated and target samples (mean ± standard deviation over 20 runs). Note: Estimated KL divergence is based on [134].

Model	2D	4D	10D
		MMD^2	
IW-CE	0.0171 ± 0.0029	0.0214 ± 0.0030	0.0214 ± 0.0044
MIW-CE	0.0246 ± 0.0038	0.0293 ± 0.0066	0.0233 ± 0.0036
SNIW-CE	0.0165 ± 0.0015	0.0197 ± 0.0035	0.0186 ± 0.0035
ID-CE	0.0304 ± 0.0025	0.0230 ± 0.0019	0.0154 ± 0.0017
IW-MMD	0.0199 ± 0.0019	0.0174 ± 0.0010	**0.0105 ± 0.0003**
MIW-MMD	0.0586 ± 0.0038	0.0342 ± 0.0016	0.0136 ± 0.0006
SNIW-MMD	**0.0149 ± 0.0011**	**0.0137 ± 0.0007**	0.0107 ± 0.0002
C-GAN	0.0174 ± 0.0040	0.0177 ± 0.0029	0.0630 ± 0.0302
		Energy	
IW-CE	0.0141 ± 0.0027	0.0361 ± 0.0044	0.0794 ± 0.0203
MIW-CE	0.0230 ± 0.0041	0.0473 ± 0.0083	0.1040 ± 0.0188
SNIW-CE	0.0144 ± 0.0037	0.0350 ± 0.0052	0.0720 ± 0.0080
ID-CE	0.0361 ± 0.0048	0.0600 ± 0.0073	0.0998 ± 0.0156
IW-MMD	0.0179 ± 0.0031	0.0341 ± 0.0120	0.0700 ± 0.0274
MIW-MMD	0.0881 ± 0.0303	0.0908 ± 0.0238	0.2123 ± 0.0893
SNIW-MMD	**0.0136 ± 0.0020**	**0.0291 ± 0.0055**	**0.0506 ± 0.0147**
C-GAN	0.0140 ± 0.0057	0.0297 ± 0.0110	0.5828 ± 0.5416
		KL	
IW-CE	0.1768 ± 0.0635	0.4934 ± 0.1238	2.7945 ± 0.5966
MIW-CE	0.3265 ± 0.1071	0.6251 ± 0.1343	3.3093 ± 0.7179
SNIW-CE	0.0925 ± 0.0272	0.3864 ± 0.1478	2.3060 ± 0.6915
ID-CE	0.1526 ± 0.0332	0.3444 ± 0.0766	1.4128 ± 0.3288
IW-MMD	**0.0343 ± 0.0230**	**0.0037 ± 0.0489**	**0.5133 ± 0.1718**
MIW-MMD	0.2698 ± 0.0618	0.0939 ± 0.0522	0.8501 ± 0.3271
SNIW-MMD	0.0451 ± 0.0132	0.1435 ± 0.0377	0.6623 ± 0.0918
C-GAN	0.0879 ± 0.0405	0.3108 ± 0.0982	6.9016 ± 2.8406

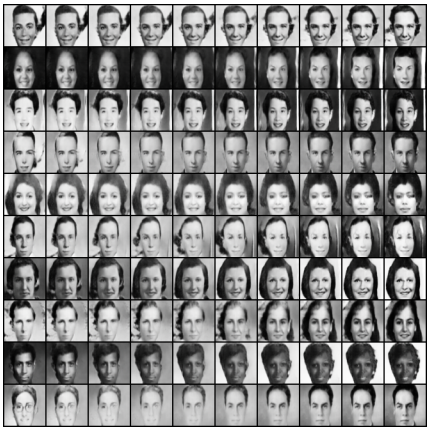
Table A.2: Squared MMD, energy distance, and estimated KL divergence between generated and target samples (best over 20 runs). Note: Estimated KL divergence is based on [134].

Model	2D	4D	10D
		MMD^2	
IW-CE	0.0140	0.0175	0.0148
MIW-CE	0.0187	0.0213	0.0157
SNIW-CE	0.0141	0.0152	0.0138
ID-CE	0.0257	0.0198	0.0128
IW-MMD	0.0172	0.0147	**0.0099**
MIW-MMD	0.0522	0.0321	0.0124
SNIW-MMD	0.0130	**0.0125**	0.0104
C-GAN	**0.0101**	0.0133	0.0152
		Energy	
IW-CE	0.0099	0.0281	0.0520
MIW-CE	0.0163	0.0331	0.0659
SNIW-CE	0.0075	0.0239	0.0584
ID-CE	0.0306	0.0476	0.0715
IW-MMD	0.0128	0.0163	0.0294
MIW-MMD	0.0570	0.0578	0.0824
SNIW-MMD	0.0107	0.0220	**0.0290**
C-GAN	**0.0061**	**0.0155**	0.0872
		KL	
IW-CE	0.0754	0.3543	1.4763
MIW-CE	0.1534	0.4110	1.9377
SNIW-CE	0.0378	0.1787	1.2751
ID-CE	0.088	0.2257	0.8249
IW-MMD	**-0.0079**	**-0.0632**	**0.1122**
MIW-MMD	0.2025	0.0171	0.2811
SNIW-MMD	0.0297	0.0733	0.4911
C-GAN	-0.0043	0.1384	1.5569



(a) Conditional DCGAN

(b) Importance Duplication

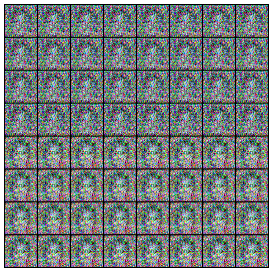


(c) Importance Weighting (IW-MMD)

(d) Median of Means (MIW-MMD)

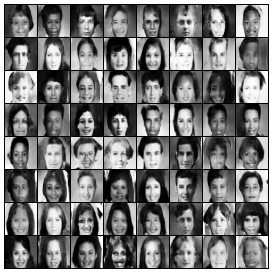
Figure A.1: Example interpolations in the latent z space, half-decades experiment.

Analogous to the self-normalized example of Section 3.3.3, we use our self-normalized estimator to manipulate the distribution over twos from the MNIST dataset, where we aim to have fewer curly twos and more twos with a flat bottom. As before, 200 were manually labeled with weights. Fig. A.5a shows 100 real images, sorted in terms of their inferred weight. Fig. A.5b shows 100 generated simulations, sorted in the same manner, clearly showing a decrease in the proportion of curly twos. Fig. 3.4c shows the inferred weights



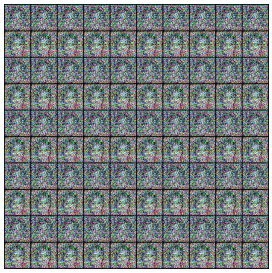
(a) Conditional DCGAN

(b) Importance Duplication



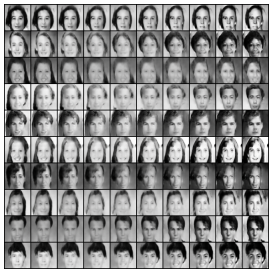
(c) Importance Weighting

(d) Median of Means

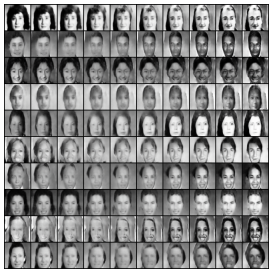


(e) Conditional DCGAN

(f) Importance Duplication



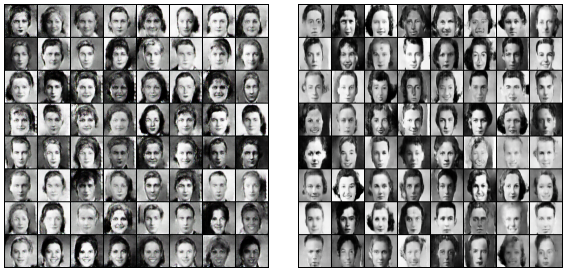
(g) Importance Weighting



(h) Median of Means

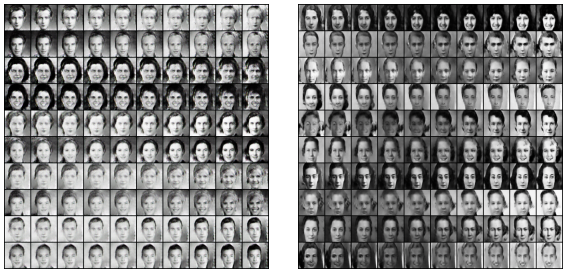
Figure A.2: Example generated yearbook images from two time periods: Old (1930) and Recent (1980-2013). The target distribution is 50%/50%, while the training set is 1%/99%. Again, C-DCGAN is unstable across a variety of training parameters, while the importance weighted MMD-GAN methods produce reasonable samples (b)–(d) with meaningful interpolations in the latent space (f)–(h).

for both real and simulated data.

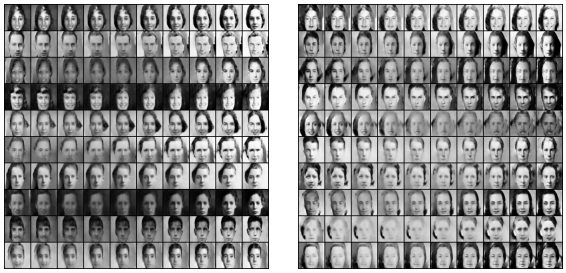


(a) Conditional DCGAN (b) Importance Duplication

(c) Importance Weighting (d) Median of Means

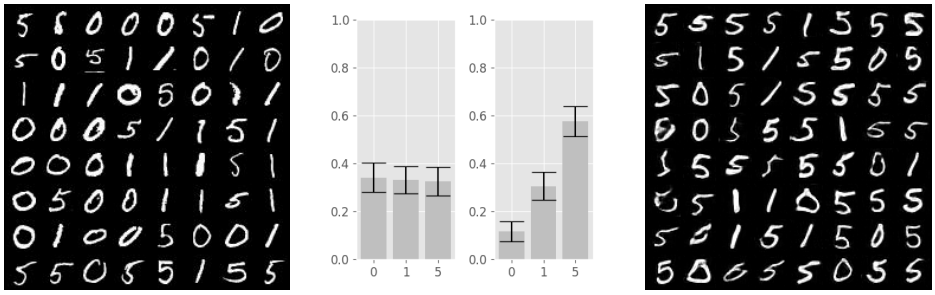


(e) Conditional DCGAN (f) Importance Duplication



(g) Importance Weighting (h) Median of Means

Figure A.3: Example generated yearbook images from two time periods: Old (1925-1944) and Recent (2000-2013). Target distribution is 83%/17% while the given data $M\mathbb{P}$ is split 50%/50%. Each time period contains enough images to train C-CDGAN successfully. However, the other methods produce qualitatively sharper images (a)–(d) with smoother latent interpolations (e)–(h).



(a) Source, even distribution of 0s, 1s, and 5s

(b) Source (left), simulation (right); target of 10%-30%-60%

(c) Simulations, boosted distribution

Figure A.4: Importance weights are used to accurately boost an even class distribution to specified levels.

(a) Data

(b) Generator

(c) KS distance

Figure A.5: A small set of labels are used to train an importance weighted estimator that aims to boost the presence of flat-bottomed twos. In A.5a and A.5b, samples are sorted by predicted weight, and in A.5c, the empirical CDFs of data, generated, and importance duplicated draws, are shown, where the latter serves as a theoretical target. The generated distribution produces more flat-bottomed twos, and is close in distance to the target, with $d_{KS} = 0.07$, $p = 0.376$.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[2] John M Abowd and Lars Vilhuber. How protective are synthetic data? In *International Conference on Privacy in Statistical Databases*, pages 239–246. Springer, 2008.

[3] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[4] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *ACM symposium on Theory of Computing*, pages 20–29. ACM, 1996.

[5] David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 156–163. ACM, 1991.

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *ICML*, 2017.

[7] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[9] Hassan Jameel Asghar, Ming Ding, Thierry Rakotoarivelo, Sirine Mrabet, and Mohamed Ali Kaafar. Differentially private release of high-dimensional datasets using the gaussian copula. *arXiv preprint arXiv:1902.01499*, 2019.

[10] Matej Balog, Ilya Tolstikhin, and Bernhard Schölkopf. Differentially private database release via kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 414–422. PMLR, 2018.

[11] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282. ACM, 2007.

[12] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[13] M.G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv:1705.10743*, 2017.

[14] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[15] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[16] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[17] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: Boundary equilibrium generative adversarial networks. *arXiv:1703.10717*, 2017.

[18] M. Bińkowski, D.J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

[19] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

[20] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[21] T. Bolukbasi, K.-W. Chang, J.Y. Zou, V. Saligrama, and A.T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NIPS*, 2016.

[22] Claire McKay Bowen and Fang Liu. Comparative study of differentially private data synthesis methods. *arXiv preprint arXiv:1602.01063*, 2016.

[23] A. Caliskan, J.J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[24] D Chaudhuri, CA Murthy, and BB Chaudhuri. Finding a subset of representative points in a data set. *IEEE transactions on systems, man, and cybernetics*, 24(9):1416–1424, 1994.

[25] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[26] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[27] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, and Hoeteck Wee. Toward privacy in public databases. In *Theory of Cryptography Conference*, pages 363–385. Springer, 2005.

[28] T. Che, Y. Li, A.P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In *ICLR*, 2017.

[29] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 286–305, Boston, Massachusetts, 18–19 Aug 2017. PMLR.

[30] Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[31] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, and Thanh TL Tran. Differentially private summaries for sparse data. In *Proceedings of the 15th International Conference on Database Theory*, pages 299–311. ACM, 2012.

[32] Botos Csaba, Adnane Boukhayma, Viveka Kulharia, András Horváth, and Philip HS Torr. Domain partitioning network. *arXiv preprint arXiv:1902.08134*, 2019.

[33] Hal Daumé III, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59. Association for Computational Linguistics, 2010.

[34] Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Automated inference with adaptive batches. In *AISTATS*, pages 1504–1513, 2017.

[35] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[36] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[37] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *STOC*, volume 9, pages 371–380, 2009.

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

[39] 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.

[40] G.K. Dziugaite, D.M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[41] C. Esteban, S.L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv:1706.02633*, 2017.

[42] Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 361–370. ACM, 2009.

[43] Kenji Fukumizu, Arthur Gretton, Gert R Lanckriet, Bernhard Schölkopf, and Bharath K Sriperumbudur. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *Advances in neural information processing systems*, pages 1750–1758, 2009.

[44] Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in neural information processing systems*, pages 489–496, 2008.

[45] Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

[46] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015.

[47] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[48] S. Ginosar, K. Rakelly, S. M. Sachs, B. Yin, C. Lee, P. Krähenbühl, and A. A. Efros. A century of portraits: A visual historical record of American high school yearbooks. *IEEE Transactions on Computational Imaging*, 3(3):421–431, Sept 2017.

[49] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[51] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 13(Mar):723–773, 2012.

[52] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[53] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213, 2012.

[54] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Random differential privacy. *arXiv preprint arXiv:1112.2680*, 2011.

[55] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.

[56] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[57] R Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*, 2017.

[58] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The annals of mathematical statistics*, pages 293–325, 1948.

[59] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[60] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[61] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[62] D.G. Horvitz and D.J. Thompson. A generalization of sampling without replacement from a finite universe. *JASA*, 47(260):663–685, 1952.

[63] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.

[64] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[65] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[66] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, and A. Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883, 2017.

[67] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

[68] Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348, 1992.

[69] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[70] Herbert KH Lee, Matthew Taddy, and Genetha A Gray. Selection of a representative sample. *Journal of classification*, 27(1):41–53, 2010.

[71] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[72] Jaewoo Lee and Chris Clifton. Differential identifiability. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1041–1049. ACM, 2012.

[73] David Leoni. Non-interactive differential privacy: a survey. In *Proceedings of the First International Workshop on Open Data*, pages 40–52. ACM, 2012.

[74] Matthieu Lerasle, Zoltán Szabó, Timothée Mathieu, and Guillaume Lecué. Monk–outlier-robust mean embedding estimation by median-of-means. *arXiv preprint arXiv:1802.04784*, 2018.

[75] Christophe Ley, Gesine Reinert, Yvik Swan, et al. Steins method for comparison of univariate distributions. *Probability Surveys*, 14:1–52, 2017.

[76] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, 2017.

[77] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, 2017.

[78] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 123–134. ACM, 2010.

[79] Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Advances in database technology: proceedings. International Conference on Extending Database Technology*, volume 2014, page 475. NIH Public Access, 2014.

[80] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[81] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *ICML*, 2015.

[82] Yang D Li, Zhenjie Zhang, Marianne Winslett, and Yin Yang. Compressive mechanism: Utilizing sparse representation in differential privacy. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 177–182. ACM, 2011.

[83] Fang Liu. Model-based differentially private data synthesis. *arXiv preprint arXiv:1606.08052*, 2016.

[84] Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pages 276–284, 2016.

[85] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2370–2378, 2016.

[86] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

[87] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 57–68. IEEE, 2006.

[88] Russell Lyons et al. Distance covariance in metric spaces. *The Annals of Probability*, 41(5):3284–3305, 2013.

[89] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. \ell-diversity: Privacy beyond\kappa-anonymity. In *null*, page 24. IEEE, 2006.

[90] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.

[91] Simon Mak, V Roshan Joseph, et al. Support points. *The Annals of Statistics*, 46(6A):2562–2592, 2018.

[92] M.A. Mansournia and D.G. Altman. Inverse probability weighting. *BMJ*, 352:i189, 2016.

[93] David McClure and Jerome P Reiter. Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data. *Trans. Data Privacy*, 5(3):535–552, 2012.

[94] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[95] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–103, 2007.

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

[97] M. Mehdi and S. Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.

[98] R v Mises. On the asymptotic distribution of differentiable statistical functions. *The Annals of Mathematical Statistics*, 18(3):309–348, 1947.

[99] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[100] Arkadii Nemirovskii, David Borisovich Yudin, and Edgar Ronald Dawson. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[101] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 665–676. ACM, 2007.

[102] nhartland. Kl divergence estimators. `https://github.com/nhartland/KL-divergence-estimators`, 2018.

[103] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

[104] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

[105] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017.

[106] A.B. Owen. Monte Carlo theory, methods and examples. Book draft, 2013.

[107] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[108] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[109] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

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

[111] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[112] Anant Raj, Ho Chung Leon Law, Dino Sejdinovic, and Mijung Park. A differentially private kernel two-sample test. *arXiv preprint arXiv:1808.00380*, 2018.

[113] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[114] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2 edition, 2004.

[115] Benjamin IP Rubinstein and Francesco Alda. Pain-free random differential privacy with sensitivity sampling. *arXiv preprint arXiv:1706.02562*, 2017.

[116] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[117] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NIPS*, 2016.

[118] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[119] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

[120] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[121] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. Non-parametric estimation of integral probability metrics. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1428–1432. IEEE, 2010.

[122] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*, pages 26–37. ACM, 2016.

[123] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

[124] D.J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.

[125] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671:1–34, 2000.

[126] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[127] Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272, 2013.

[128] Martin Szummer and Rosalind W Picard. Indoor-outdoor image classification. In *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pages 42–51. IEEE, 1998.

[129] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[130] Zhehang Tong, Dianxi Shi, Bingzheng Yan, and Jing Wei. A review of indoor-outdoor scene classification. In *2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*. Atlantis Press, 2017.

[131] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of mcmc. *arXiv preprint arXiv:1903.08008*, 2019.

[132] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[133] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):57, 2018.

[134] Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

[135] Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pages 2493–2502, 2015.

[136] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[137] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering*, 23(8):1200–1214, 2010.

[138] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 12(12):3081–3093, 2017.

[139] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *arXiv preprint arXiv:1705.00609*, 2017.

[140] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[141] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

[142] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *ICLR*, 2017.

[143] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017.

[144] Ji Zhao and Deyu Meng. Fastmmd: Ensemble of circular discrepancy for efficient two-sample test. *Neural computation*, 27(6):1345–1372, 2015.

[145] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

Index

Abstract, vi
Acknowledgments, v
Appendices, 70

Background, 4
Bibliography, 97

Conclusion, 69

Dedication, iv

Importance Weighted Generative Networks, 19, 71
Introduction, 1

Support Points and Privacy, 41
Synthetic Medical Records, 37

Vita

Maurice Diesendruck was born in Castro Valley, California on May 20 1988, the son of Samuel L. Diesendruck and Esther A. Diesendruck. He received the Bachelor of Science degree in Mathematics and Economics from the University of California at Los Angeles. ... work history... he applied to the University of Texas at Austin for enrollment in their statistics program. He was accepted and started graduate studies in August, 2014.

Permanent address: 4903 Avenue F, Unit B, Austin, TX 78751

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.