HUMAN-CENTRIC DEMAND-SIDE MANAGEMENT: REVEALING LIFESTYLES, PRESERVING PRIVACY, AND PROMOTING FAIRNESS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Xiao Chen
January 2022

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Ram Rajagopal) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Abbas El Gamal)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Martin Fischer)

Approved for the Stanford University Committee on Graduate Studies

Abstract

Power grids are shifting from fossil fuel generation towards renewables such as solar and wind, driven by decarbonization targets. Because of variability and uncertainty in generating power from renewables, we face new challenges of balancing supply and demand in the power grid. To address these challenges, we investigate demand-side management (DSM) because it can be cheaper to run compared with reserving the traditional backup generation or procuring energy in the real-time market. However, some key issues are overlooked in the current DSM like understanding user behavior, preserving user privacy, and preventing discrimination against certain users such as those less able to carry out DSM or respond to pricing (e.g., time-of-use price). This dissertation primarily focuses on demand-side management in electricity systems and presents scalable frameworks to gain insights from household electricity data, to protect private attributes associated with the electricity data, and to promote fairness in managing demand-side services.

We obtain household behavioral insights from residential meter data by introducing a new concept–dynamic energy lifestyles–that characterizes behavioral patterns of household energy use in different temporal spans. We also introduce new metrics and machine learning approaches in the context of energy data analysis, both of which are needed to obtain a meaningful number of energy lifestyles. These lifestyles help us to better understand both stability and change patterns of a household's energy use over time. Our approach and results can be used by utility companies or energy service providers for identifying households to install rooftop solar and differentiating households' demand flexibility to promote dynamic pricing.

To address privacy issues specific to the energy domain, we build a framework

that preserves data quality and protects sensitive information. Privacy is quantified by the correlation between sensitive attributes (e.g., income) and the data we need to use. Taking into account the tradeoff between data privacy and data utility and inspired by generative adversarial networks (GAN), we formulate a data sharing task as a game between a data actuator and an adversarial user who aims to infer the sensitive information, then use minimax optimization to alter the raw data. Our results indicate that privacy can be preserved with limited performance loss (5%–12%) on data utility tasks.

To tackle the challenge of ensuring fairness in DSM, we investigate a use case: engaging users in demand response programs. In this case, privacy restriction must be relaxed, because fairness cannot be obtained by blindness to the protected attribute (e.g., race, income, etc.). We propose a general form of stochastic optimization that treats different groups similarly via fairness constraints in light of uncertain electricity demand. Moreover, when a limited set of demand reductions are revealed, we cast the stochastic optimization into the multi-armed bandit setting and introduce new methods to solve it with sublinear regrets.

Overall, we propose conceptual frameworks and develop new methods, all of which are operationalized with data and have the ability to advance human-centric demand-side management. Such an impact can help utility operators to plan and provide new energy services.

Acknowledgments

I am extremely fortunate to have been advised by Professor Ram Rajagopal throughout my graduate study and research. His constant support and insightful advice are what made this thesis possible. I owe so much of my accomplishment to Ram in that he taught me the right questions to ask in research, trained me how to tackle a research problem, and provided me a perfect balance between guidance and freedom. As an incredible mentor, Ram is always there to guide me along the way that helps me to think problems in different depths and breadths. Beside the academic inspirations, Ram has given indispensable life advice as I went through ups and downs during my graduate study. I couldn't have got this far on the path of PhD study without his encouragement and support.

I would like to thank my reading committee members Professor Abbas El Gamal and Professor Martin Fisher. I feel very grateful to have Abbas as my reader, as I am impressed by his sharp insights on many different engineering problems and his gifted ability of reducing complex problems to their key points. I am also blessed to have Martin as my reader. I took Martin's class on AI in construction and then was inspired by his comments on having critical thinking and motivating examples over research projects. I also want to thank Professor Dorsa Sadigh and Doctor Chin-Woo Tan as my oral committee members. I am indebted to Dorsa since her generous support made my PhD defense possible. I also enjoy every discussion with Chin-Woo, who exchanges many ideas with me on not only the research problems but also many life-related topics.

I feel very grateful to collaborate with many talented people during my graduate study. Especially, I thank Thomas Navidi, Dr. Peter Kairouz, Dr. Chong Huang,

Prof. Lalitha Sankar, Dr. Chad Zanocco, Dr. June Flora, Prof. Hilary Boudet, Sila Kiliccote, Dr. Emre Can Kara, Prof. Stefano Ermon, Prof. Marshall Burke, Prof. David Lobell, Prof. Pascaline Dupas, Prof. Jeremy Weinstein, Todd Stiers, Dr. Christopher Flores, Dr. Robert Kavaler, Floyd Williams, Prof. Hao Wang, Dr. Nicolay Laptev, Dr. Yuting Ji, Prof. Jianxiao Wang, Dr. Hao Sheng, Prof. Andrew Ng, for being my coauthors on different projects. Working with these people shaped my research views and enhanced my skills on solving various technical challenges.

Over the past few years in Stanford, I had the precious opportunity to work with many fellow students and scholars. Communicating and interacting with gifted people is a constant source of learning and a major reason why I enjoy research. I notably thank my former labmates Dr. Raffi Sevlian, Dr. Jungsuk Kwac, Prof. Junjie Qin, Dr. Jiafan Yu, Prof. Yang Yu, Dr. Yizheng Liao, Dr. Sid Petal, Dr. Sam Borgeson, Dr. Michaelangelo Tabhone, Prof. Baosen Zhang, Prof. Yang Weng, Prof. Wenyuan Tang, Dr. Mohammad Rasouli, Dr. Heidi von Korff. Their enormous capabilities and deep thoughts expanded my research vision and enriched my personal life. I also thank previous scholars Prof. Subramanian Ramamoorthy, Prof. Nicolas Astie, and Dr. Marie-Louise Arlt for the interesting discussions on the topics of control, optimization, and economics. In addition, I would like to thank current lab members and friends Gustavo Cezar, Aaron Goldin, Jose Bolorinos, Lily Buechler, Siobhan Powell, Ryan Triolo, Justin Luke, Oskar Triebe, Tao Sun, Sonia Martin, Anthony Degleris, Zhecheng Wang, Tianyuan Huang, Tina Diao, and Ye Ye. Beside working on various research projects, I would like to express my gratitude to Kelly Harrison who helped me to improve my technical writing significantly over the past few years.

Finally, I feel thankful for Prof. Zhen (Sean) Qian who brought me to the world of academic research. I wouldn't have gone along this PhD path without his help. Moreover, I would like to thank my parents on the other side of the pacific ocean, for their love and support during my graduate study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The rapid growth of computation capability, the massive integration of sensing technologies, and the increased connectivity of networks in recent decades have created new opportunities to uncover various patterns and help us make decisions in our daily lives from individual-level (online shopping promotions, video or music recommendations, and mapping or navigation services) to group-level (clinical trials, natural disaster responses, and public health policies with respect to pandemics control, e.g., the COVID-19 "shelter-in-place" and/or "reopen" decisions). All of these emerging services, products, and decision processes often produce a massive amount of data that is, in turn, being used to affect our everyday decision-making by people who closely engage with large-scale systems such as energy, transportation, and healthcare. As a result of having a lot data that was collected from heterogeneous sources, in different forms, and with various qualities, being able to process, analyze, and leverage data effectively has become a crucial challenge.

One specific system that has extensive interactions with humans is demand-side management (DSM) in the electricity sector. Driven by decarbonization targets and global warming alleviation, current power grids are shifting from fossil fuel generation towards renewables such as solar and wind. But the power supply generated from the wind and solar is poorly correlated with the electricity demand temporally, which

may bring extra cost of the grid for reserving back-up generators which are carbon-intensive or purchasing power from real-time market which is expensive. To address these challenges of imbalanced supply and demand, demand-side management is a promising approach because it can be cheaper to run compared with reserving the traditional backup generation or procuring energy in the real-time market. The notion of demand-side management consists of the planning, implementing, and monitoring activities of electric utilities, all of which are designed to encourage consumers to modify their level and pattern of electricity usage. We view this DSM system as having a huge potential to provide many services that closely relate to our energy use, such as energy efficiency or demand response programs, based on the high-resolution data streams from smart meters.

Constructing energy lifestyles

The modern electricity system faces numerous challenges such as satisfying the increasing load demand and integrating more renewable resources. To maintain a secure, sustainable, and resilient grid, utility companies and regulation authorities consider making use of demand-side management, which becomes promising because of several advantages over the supply side: i) the variability of a large number of small loads is likely to be less than that of a small number of large generators [1]; ii) demand loads can often respond to the grid operator requests instantaneously [1]; iii) the level of spatial and temporal flexibility that demand loads provide to the power system could be used to support the growth of renewable resources [2]. Despite seeing many benefits in managing the demand-side, providing such a service (e.g., demand response or energy efficiency programs), however, cannot ignore a key issue; that is, the load management must align with the users' or households' realistic behaviors. Without users' acceptance, loads cannot be counted into demand-side management solutions. Therefore, our important beginning step is to understand the current users' behavioral patterns of energy consumption for designing future services such as energy efficiency or demand response. More precisely, given the high-granular meter data, we ask:

Question 1 What simple and clear insights of energy consumption patterns can we

draw from datasets?

To answer this question, we introduce a new concept named energy lifestyles to characterize the behavioral patterns of household energy use in light of different temporal spans (e.g., seasons). In the context of energy data analysis, we also introduce a few new metrics and machine learning approaches both of which are needed to obtain a small handful of typical energy lifestyles. Such a meaningful representation of lifestyles helps us to understand the stability or change pattern for household energy use. This comprehensive view of energy lifestyles helps utility companies, policy makers, or energy service providers for identifying households to install behind-meter resources (e.g. rooftop solar and battery storage) and differentiating households' demand flexibility to promote dynamic pricing.

Indeed, ubiquitous data can provide many benefits in applications. However, such actuations of data, which give rise to an abundance of ways to collect and publish personal information, bring up a concern about privacy risks. For example, National Institute of Standards and Technology (NIST) and U.S. Department of Energy (DOE) have issued guidelines for ensuring privacy and security of the smart grid and other critical infrastructure system [3, 4]. However, some specific policies are not easy to implement, especially when it comes to privacy, because the privacy is subjective in its nature: privacy is interpreted and quantified in various ways by different people and in different contexts. On the other hand, data that are considered private, or that have the potential to reveal sensitive information about individuals (e.g., gender, house sq-ft, income, etc.) are necessary to obtain insights or draw conclusions in the aggregate. As we can see, there is a natural tradeoff between data privacy and data utility (usefulness of the data). Managing this tradeoff is a key challenge in developing data-driven methods for our lives, since this privacy-utility tradeoff is becoming more human-centric in that we, as users, are actively participating in the infrastructure systems. Failure to protect privacy may result in significant harm to individuals and to our society.

Data privacy

In 2006, Netflix, a DVD-rental and video streaming service, opened a competition with a one million dollar prize to whoever could improve their recommendation system by 10% based on their released dataset, which contained anonymous movie ratings of 500,000 subscribers. By matching users' movie ratings with reviews given by them on IMDB, a separate large online movie database, Narayanan and Shmatikov [5] showed how it was possible to identify "anonymous" users in the dataset. After the Federal Trade Commission raised privacy concerns that resulted from this study, a second Netflix challenge was cancelled.

The problem of privacy leakage is not restricted to online streaming services. In the energy sector, the demand-side management system also has privacy issues with respect to analyzing smart meter data. For example, Beckel et al. [6] found that fine-grained electricity consumption data can lead to identifying specific characteristics that may reveal information about a home's socio-economic status, dwelling, and appliances, with an accuracy of more than 70% for all households. Given the concern with privacy, it is important to understand possible approaches to maintaining digital privacy for customer data associated with smart meters. The central question in privacy-preserving data analysis that we would like to ask:

Question 2 How can we release a dataset or accurate statistics about a dataset while protecting the privacy of those individuals who contributed the data in demand-side management ?

To tackle this question, we provide a privacy preserving framework that preserves data quality while protect sensitive information. The privacy here is quantified by correlation between sensitive attributes (e.g., gender or income) and data we try to exploit. Such a quantification of privacy stems from the notion of information leakage measured by information-theoretic approaches. Based on our intuition of the tradeoff between data privacy and data utility, we formulate releasing the data as a game between a data actuator and an adversarial user who aims to infer the sensitive information, and then we use a generative adversarial network (GAN) or stochastic optimization to alter the raw data in a favorable way.

Not all problems arising in the presence of sensitive datasets are a matter of privacy. Because many data–driven applications consist of high–stakes decision-making models, these models are required to be fair with respect to protected classes from ethical and legal perspectives. Examples of these decision–making problems include employment decisions, high school or college admissions, credit loan approvals, and so on. For instance, in finance, discrimination on the basis of certain protected classes, such as gender, race, religion, is prohibited per the Fair Housing Act (FHA) [7] and the Equal Credit Opportunity Act (ECOA) [8]. Because data generated by the underlying systems sometimes may not reflect the entire society, making decisions based on a skewed portion of those data will likely reflect the biases against certain groups, and those decisions in turn exacerbate the systemic biases against particular groups.

Fairness in decision making

COMPAS, an acronym for Correctional Offender Management Profiling for Alternative Sanctions, is a software-and-support tool used to predict recidivism risk, the risk that a criminal defendant will re-offend. Researchers from ProPublica showed that the COMPAS algorithm was biased in determining black convicts to be at a higher risk of recidivism [9], that is, Black people are almost twice as likely as whites to be labeled higher risk even though the rates of re-offending are approximately the same.

This case of recidivism is just one example of systematic bias against certain groups of people. The underlying problem here is not restricted to the jury and bail system. Such biased decisions or unfair consequences of decisions also exist in the energy sector. For example in demand-side management, the time-of-use (TOU) rates may disproportionately impact energy bills and health of vulnerable households that face greater energy needs combined with greater social and financial pressures [10]. A recent example is that COVID-19 pandemic is expected to increase the prevalence of energy poverty especially among African American households because of a history of segregating housing policy and structural deficiencies of homes [11]. Broadly speaking, the question we ask here is:

Question 3 How can we prevent discrimination against certain subgroups or individuals throughout the decision-making processes in demand-side management programs?

To analyze this question, we investigate a use case in the context of energy services: engaging users in demand response programs. We argue that fairness cannot be obtained by blindness to the attribute we want to protect, especially in the case of the group fairness. We propose a general online optimization that captures the fairness constraints in light of the uncertain electricity demand. Moreover, when a limited set of demand reduction is revealed, we cast the traditional stochastic optimization into a multi-armed bandit setting and introduce new methods to solve the optimization problem with bounded regrets.

In short, this thesis is a first step in answering the previous three questions in the context of demand-side data analysis, data sharing, and program decision-making. We summarize major contributions in the following section and expand more details in the subsequent chapters.

1.2 Thesis Overview and Contributions

In Chapter 2, we propose a new framework for understanding residential electricity demand by using a dynamic energy lifestyles approach that is iterative and highly extensible in response to Question 1. To obtain energy lifestyles, we develop a novel approach that applies Latent Dirichlet Allocation (LDA) to extract a series of latent household energy attributes. By doing so, we provide a new perspective on household electricity consumption where each household is characterized by a mixture of energy attributes that form the building blocks for identifying a sparse collection of energy lifestyles. We then use clustering techniques to derive six distinct energy lifestyle profiles, namely *active mornings, night owl, everyday is a new day, home early, home for dinner*, and *steady going*, which are highly interpretable in terms of daily activities. Our lifestyle approach is also flexible to varying time interval lengths, and we provide an example of our lifestyle approach applied seasonally to track energy lifestyle dynamics within and across households. These energy lifestyles are then compared

to different energy use characteristics, and we discuss their practical applications for energy program design and lifestyle change analysis. The results in this chapter are synthesized from the joint work with Chad Zanocco, June Flora, and Ram Rajagopal, which are available in [12].

Next, we use Chapter 3 and Chapter 4 to cover the Question 2. In Chapter 3, we introduce a decoupling of the creation of a latent representation and the privatization of data that allows the privatization to occur in a setting with limited computation and minimal disturbance on the utility of the data. We leverage a generative linear model to create a privatized representation of the data and establish the connection between this simple linear transformation of generative noise with differential privacy. We also run empirical experiments on realistic high-dimensional datasets with comparison to the related studies. Additionally, we build a connection between solving constrained optimization and having Rényi differential privacy under certain conditions. Finally, we improve the Autoencoder to have a robust decoder for reconstructing perturbed data, and then draw comprehensive insights on: (i) the latent geometry of the data distribution before and after privatization, (ii) how training against a cross-entropy loss adversary impacts the mutual information between the data and the private label, and (iii) how our linear filter affects the classification accuracy of sensitive labels. The results in this chapter are partially selected from the joint work with Thomas Navidi, Peter Kariouz, Chong Huang, Stefano Ermon, Lalitha Sankar, and Ram Rajagopal, and are published in [13, 14, 15].

In Chapter 4, we propose a minimax approach to generate realistic meter data that is decorrelated from sensitive attributes while maintaining limited performance loss of a cost minimization optimal control algorithm using battery storage. Additionally, we developed a parallelized method that can be easily incorporated in modern deep learning architectures. The correlation of data privatized by our method with sensitive attributes and the performance of a control algorithm is evaluated on two real datasets of residential power demand: one with synthetic sensitive labels and one with real labels. We demonstrate that our method is able to decrease the classification accuracy of an adversary by over 20% while maintaining the performance of the optimization to within 10% over both datasets. This chapter is joint work with Thomas Navidi

and Ram Rajagopal, published as [16].

Motivated by the concerns of fairness in decision making process with regards to Question 3, we aim to find a fair approach in model learning and optimization in the context of energy services, more specifically, the demand response program in Chapter 5. We propose two types of fairness metrics, i.e., group fairness and individual fairness, and explicitly consider fairness constraints in the DR optimization problem. We develop two online learning algorithms for the fairness-aware customer selections. The first algorithm is an upper-confidence-bound approach that adaptively solves a constrained convex optimization. The second algorithm is a primal-dual decomposition-based approach that uses mirror descent to find fair selections of participants. We prove that both algorithms achieve sub-linear regrets compared to the offline fair selections with the full knowledge of customers' load reduction. Finally, we conduct numerical experiments based on both synthetic and real load data, which demonstrate that both algorithms can effectively solve the DR optimization problem by optimally curtailing loads while selecting customers in a fair manner. This chapter is joint work with Hao Wang and Ram Rajagopal.

Chapter 6 makes concluding remarks and raise additional questions for future researches about data analytics with privacy and fairness in the energy domain.

Chapter 2

Constructing Energy Lifestyles using Latent Dirichlet Allocation

The rapid expansion of Advanced Meter Infrastructure (AMI) has dramatically altered the energy information landscape. However, our ability to use this information to generate actionable insights about residential electricity demand remains limited. In this chapter, we propose a new framework for understanding residential electricity demand by using a dynamic energy lifestyles approach that is iterative and highly extensible.

2.1 Introduction

The growth of advanced metering infrastructure (AMI) has greatly expanded our potential to analyze household electricity usage. To date, AMI infrastructure provides hourly and sub-hourly electricity usage patterns via smart meter technology for tens of millions of households in the United States alone, with the deployment of residential smart meters increasing yearly by millions [17]. Prior analysis of smart meter data has provided insights about household daily load shapes and the variation of electricity use patterns both within and across households. While such information about household energy use patterns is being applied toward forecasting residential demand [18, 19, 20], other nascent applications of smart meter data are

of increasing interest to energy providers, including targeting households for demand response (DR) [21, 22, 23], tailoring information to differing user segments about energy efficiency (EE) programs [24], and making recommendations to customers for enrolling them into pricing programs, such as time–of–use rate plans [25, 26]. Moreover, in many extant applications, household energy use pattern information is typically treated as static, without consideration into how patterns may change across time, either cyclically (e.g., seasons, school calendar year, etc.) or as structural household shifts (e.g., new household members, change in work hours, etc.), potentially missing opportunities for more refined targeting, tailoring, and other program design considerations.

As smart meters become more ubiquitous in households across the world, new advances are needed to process the deluge of data streams produced from these devices and then generate actionable insights—especially in a way that has small computational overhead and does not require continuous human–in–the loop interactions [27]. In particular, using smart meter data to understand household level energy use is an on-going challenge, and with it comes the difficulties in developing meaningful interventions that can ease burdens on the grid while also contributing to energy system decarbonization and maintaining customer engagement and satisfaction [28]. For example, one such motivation for household–level energy interventions is to reduce greenhouse gas emissions from nonrenewable generation sources (e.g., natural gas "peaker" plants) during periods of high demand, while also expanding the potential for customers to change their energy behaviors and appliance purchases to save money on their energy bills [29].

While existing work on customer identification and segmentation has been explored in the literature [22, 30, 21, 26], insights about customer segmentation for households and groups of households do not demonstrate strong linkages to a variety of common occupant behaviors with few exceptions [31] (see section 2.2 for examples). Additionally, while many methods have been proposed for gaining broad insights into customers and groups of customers' electricity consumption, these methods are often too complex to easily scale for use by utility companies or energy providers [32].

Existing methods usually require higher-resolution data than what is typically available via smart meters, and these methods have not produced the sort of broadly generalizable insights needed to effectively inform program design [33].

To address these challenges, we use Latent Dirichlet Allocation (LDA) to analyze daily energy consumption patterns of households. While LDA is most commonly associated with Natural Language Processing tasks such as extracting latent topics from text-based documents [34, 35], it is now increasingly applied in other domains and problem areas, including the remote sensing of satellite imagery [36, 37] for understanding image semantics, environmental science for interpreting policy change in dealing with climate change [38] and biology and genomics for tasks ranging from extracting common patterns of mass fragments to neutral losses in molecules [39, 36].

We propose yet another application for LDA previously unexplored: the classification and interpretation of energy use patterns in the home. In doing so, we seek to identify latent patterns of daily energy consumption and then use these latent constructs to build residential energy lifestyle profiles. Our approach does not directly characterize residential energy activities and behaviors through observational or self-reported methods [40, 41] or real time data disaggregation of household energy consumption–all of which can introduce complexity that makes it challenging to generalize across households. Instead, our conceptualization of energy lifestyles are more broadly construed, with the potential to generate meaningful insights without having to resort to finer grained, more nuanced understandings of energy use and energy–related activities within the home. Such an understanding of energy lifestyles could have applications for energy practitioners, such as electricity service providers, policymakers, and the research community for tasks including identifying energy use patterns, targeting customers, and understanding household demand flexibility and response of residential users.

Our approach toward developing energy lifestyles also affords us new opportunities in examining the temporal dimensions of lifestyles, or how these energy lifestyles may change across time intervals of varying length. Previous research has considered a lifestyle as a static attribute of a household, with the lifestyle referring to a component that does not change across time. However, research suggests that lifestyles can

indeed have dynamic components [42, 43, 44, 45, 46, 47], yet much of this literature is limited to within–day time organization as opposed to across days, weeks, months etc. On a global scale, we have recently experienced dramatic disruptive influences that has changed the nature, organization, and amount of electricity consumed–lifestyle changes that have occurred during the COVID-19 pandemic [48, 49, 50]. While the measurement of lifestyle change through electricity use may only serve as an approximation for a variety of conditions and activity patterns that occur within a household across time, we postulate that such a lifestyle approach could provide a signal for when large changes related to energy use occurs in the home. Such changes could include anything from a change in the number of household occupants (e.g., a child being born or leaving for college) to a change in the patterns of occupancy (e.g., new employment or retirement) to broader "shocks" such as recent COVID-19 related restrictions. On the other hand, some households may experience little to no change in energy lifestyles across time, also providing an important insight into the stability of energy use patterns and their associated household activities. Understanding these characteristics of lifestyles could bring new opportunities for energy providers to dynamically target energy programs during certain times throughout the year, and allow the iterative identification of lifestyle patterns based on constantly updating data streams from AMI infrastructure. While this understanding has the potential to both improve recruitment and engagement in efficiency and demand response programs [51, 23], we may also find that this more "real" life understanding of residential consumption leads to the development of new policies and programs.

In this research we break new ground in constructing temporally dynamic energy lifestyles using a novel application of LDA. Given this focus on generating and gaining insights from temporally dynamic energy lifestyles, our research seeks to answer the following research questions: (1) What residential energy lifestyle profiles emerge from empirical data and what are their prominent characteristics? and (2) What patterns of change, or stability, is observed in households' lifestyle profiles across time and what is related to these temporal dynamics?

We address these research questions in the following sections. First, we describe our approach for generating energy lifestyles by introducing our conceptual framework

and method. Next, we describe our residential electricity dataset and experimental setting. Then we derive energy lifestyles and provide insights about their prominent features and patterns of change across time. Lastly, we discuss applications of this lifestyle approach and provide recommendations for future research.

2.2 Constructing Energy Lifestyles

2.2.1 Conceptual framework

While there are many ways to describe a lifestyle, we adopt the definition that a lifestyle is the consolidation of a persistent set of patterns of behavior that occur within the home environment [52]. We propose that energy consumption is best understood as a consequence of lifestyles that reflect the organization, sequencing, synchronicity, habitualness, and contingent or interdependence of the timing of the activities of daily life within a day and over weeks, months and years.

To capture the temporal patterns in energy use, our conceptual approach to energy lifestyles is built around the daily load shape as a core feature of household consumptive patterns, which imparts information about the relative magnitude, duration, and timing of energy use throughout a day (24 hour period). Embedded within this daily load shape representation is information about energy use related to the timing of household activities (e.g., cooking, cleaning, entertainment), appliance characteristics (e.g., heating/cooling technologies), household characteristics (e.g., number of occupants, age of occupants, etc.) and contextual and environmental characteristics (e.g., weather, climate). Features of the daily load shape, including the timing of peak, base, and the ratio of peak/base, contribute to insights about the relation between activities and electricity use. Finally, the variation of load shape patterns (i.e., entropy) imparts information about consistency or inconsistency of energy use patterns across time. The load shape itself, therefore, contains rich information about a household's energy use and everything within the household related to this use.

To encompass this broad representation of household energy use with daily load shape as a focus, we envision a framework that applies Latent Dirichlet Allocation (LDA). LDA is a generative statistical model that allows unobserved groups to be

explained by a set of observations that have related characteristics. The canonical example of LDA is identifying topics in text analysis [53, 54, 39], where words are observations that are collected from documents, such as a newspaper article, and each document is some mixture of topics that can later be assigned semantic meaning (e.g., politics, sports, etc.). In the text-based example, the process of generating a document is described by a sampling of topics from a mixture of topics, and a sampling of corresponding words according to those topics, and then repeating this process to generate all words in the document. Topics are initialized randomly and then updated through iterations using Variational Bayesian Inference [34, 55] or Markov Chain Monte Carlo [56] approaches until a convergence criteria is met, where convergence is measured by the change of likelihood in producing the observations (words), or the change of the inferred parameters.

Table 2.1: Conceptual relationship between text analysis and energy analysis in a LDA setting

Text and language		Residential energy consumption
documents	$\longleftrightarrow$	households
words	$\longleftrightarrow$	load shapes
topics	$\longleftrightarrow$	energy attributes

Applying LDA to the context of analyzing energy demand, we develop a novel application that extracts latent patterns of energy consumption by considering households as documents, and treating load shapes as words. A conceptual relationship of terms in the domain of language analysis and in the case of our proposed energy analysis is shown in Table 2.1. In this comparison example, latent energy patterns, which we have named an energy attribute, is synonymous with a topic in the text analysis example.

This framework is expected to generate two nested components. The first component is the aforementioned energy attribute, a latent characterization of daily energy use patterns. These energy attributes are derived from daily load shapes and form the building blocks of dominant daily energy use patterns in a household, and each

household can contain different proportions of these latent attributes. When proportions of these energy attributes are aggregated across a large pool of households, their mixtures among certain household-types can be used to generate the second layer of abstraction which we refer to as an "energy lifestyle". Energy lifestyles, therefore, are an expression of collective daily energy use patterns across groups of households. These energy attributes and energy lifestyles can be applied to any temporally consumptive data stream (e.g., electricity, gas, and water) dependent on the availability of such information.

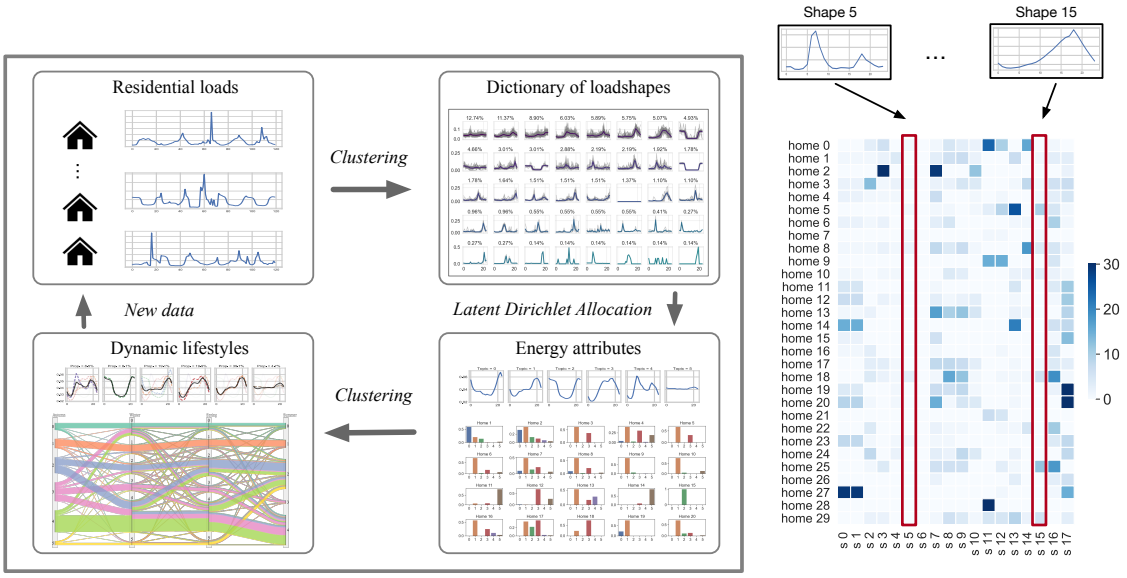
2.2.2 Overview of methods

In the context of analyzing energy demand, we develop a method to extract latent patterns using LDA. In our energy analysis case, analogous to the document example where each document contains a mixture of topics, we assume that each household contains a mixture of energy attributes. Therefore, an objective is to identify latent attributes of energy consumption across many households and construct load shapes denoted as s. Specifically, for a j-th home having a mixture of K attributes, the household's attribute mixture weights θ_j is a probability distribution drawn from a Dirichlet prior with parameter α and the k-th attribute is a multinomial distribution ψ_k over a S-shape vocabulary (or dictionary). For i-th shape s_{ji} in home j, a topic $z_{ji} = k$ is sampled from θ_j and s_{ji} is drawn from ψ_k. The generative model can therefore be expressed as

$$\theta_j \sim Dir(\alpha), \psi_k \sim Dir(\beta), \{z_{ji} = k\} \sim \theta_j, s_{ji} \sim \psi_k \quad . \tag{2.1}$$

We briefly describe the LDA method here to build intuition about its application and then we expand upon this by providing a more detailed description in 2.5.8. Once energy attributes are finalized, we then apply the k-means clustering method on the energy attribute space of households in the second stage to generate a sparse representation of energy usage patterns over days (characterized by cluster centers), which we refer to as energy lifestyles because they contain latent patterns of energy usage

generated across households. To provide an overview of the entire process of generating energy lifestyles, we constructed a simplified workflow displayed in Figure 2.1a, described in detail below.



(a) Schematic flow of generating dynamic lifestyles.

(b) Frequency counts of a few shapes

Figure 2.1: (a) The process of generating dynamic lifestyles. We first cluster raw smart meter data to create a dictionary of load shapes. Next, we apply LDA to identify representative energy attributes. Based on these attributes, we finally compose the lifestyles across time. (b) We present the frequency counts of 18 load shapes in a random subset (denoted as s 0 to s 17 on x-axis) of 30 homes (*home* 0 to *home* 29 on y-axis) over the Autumn season (Sept. - Nov.).

To dynamically create and analyze energy lifestyles, we gather a collection of daily load shapes that covers a majority of patterns of household consumption by clustering over raw meter data (Figure 2.1a). Such a collection forms a load shape dictionary that allows us to identify the frequency of generalized load shapes for each household (Figure 2.1b). After obtaining the frequency counts of shapes for households, we use the LDA method to yield representative latent energy attributes. Correspondingly, the households can also be viewed as mixtures of attributes (Figure 2.1a). Because many households share similar energy attribute distributions, we use a clustering method to group similar households in their energy attribute space, yielding distinct

clusters (i.e., lifestyles). Each cluster represents an energy lifestyle group that can be further interpreted using the proportions of attributes of energy use patterns within the cluster.

In addition, to explore temporal dynamics we apply the LDA method on seasonal intervals (Autumn, Winter, Spring, Summer) and assign each household to its nearest computed lifestyles, which we display as seasonal transitions of lifestyles for households (Figure 2.1a). We can then iterate on these steps and re-generate insights of household consumption patterns as data streams are updated. While in this research we explore temporal dynamics seasonally, this method can be applied across other time intervals (e.g., monthly, bi-annually, annually, etc.).

2.3 Experiments and Results

Our framework is heavily driven by empirical data from actual residential households using several contemporary machine learning methods. Specifically, we first run an experiment for generating a representative load shape dictionary by clustering raw residential smart meter data. In our next series of experiments, we apply this load shape dictionary and then synthesize typical lifestyles by using LDA. When summarizing the lifestyle profiles, we assign them names according to a composite shape formed via reconstructing the weighted sum of load shapes. Once these lifestyle profiles are obtained, we run a set of experiments to validate the profiles by examining the electricity consumption features and show these features (e.g., the ratio of morning to whole day energy use) support temporal characteristics of these lifestyles. Finally, we run a series of experiments to identify households that change lifestyles across seasons (Changer) and those who do not (No Changer). We describe our data and approach for analyzing this data in detail in the following sections.

2.3.1 Description of residential smart meter data

Our work utilizes a large dataset of residential load consumption from Pacific Gas & Electricity (PG&E) spanning from August 1, 2010 to August 1, 2011, which contains more than one hundred thousand customers. For analysis, we randomly selected

60,000 homes having hourly smart meter data, comprising 436 ZIP codes in California, U.S.A, covering eight different climate zones (Figure 2.12). Such a sample population, which is larger than many previous studies [57, 6], is appropriate for capturing heterogeneity in residential energy consumption patterns. From this data, we then convert each household's load shape pattern into the format of (24-hour) daily sequences over the course of a year for our analysis.

2.3.2 Dictionary of load shapes

Since households display a variety of load shapes across time, and that the mixture of these load shapes is associated with the lifestyle that households may possess, we first learn a dictionary of daily load shapes that is the foundation of our energy lifestyle approach. To generate a robust dictionary of load shapes, we utilize clustering methods with a careful selection of distance metrics (2.5.2).

Given a set $\mathcal{X}$ that includes all daily electricity loads and a data point $\boldsymbol{x} \in \mathcal{X}$, we would like to find a number of representative points of clusters, denoted as a set $C \supseteq \mathcal{X}$, that can summarize a massive dataset into several typical patterns. To accomplish this, we minimize the distance between points and cluster sets $\min_{\boldsymbol{x}\in\mathcal{X}} d(\boldsymbol{x}, C)$ in metric d, where $d(\boldsymbol{x}, C) = \min_{c\in C} d(\boldsymbol{x}, c)$ is the minimum distance from $\boldsymbol{x}$ to a center c. Taking the standard k-means as an example, we have an assignment $\phi : \mathcal{X} \to C$ of points to clusters so as to $\min_{\boldsymbol{x}\in C} d(\boldsymbol{x}, \phi(\boldsymbol{x}, C)) = \min_C \sum_{i=1}^{k} \min_{\boldsymbol{x}\in C_i} ||\boldsymbol{x} - c_i||_2^2$, where d is the Euclidean distance between two points. Besides the Euclidean distance, we also apply the cosine distance, the L^1 distance, and the dynamic time warping (DTW) distances [58] to perform the clustering for the load shape dictionary. We also test k-medians [59], hierarchical clustering [60], and DBSCAN [61] clustering methods for comparison (see 2.5.2 for more information). We set the load shape dictionary of size 200 using the k-medians method with a hybrid of DTW and Euclidean distances, because this setting yields a good coverage of profiled shapes with the highest score on the Calinski-Harabaz Index [62]. Further explanation is provided in 2.5.2

2.3.3 Energy lifestyles composition

Once we have derived a dictionary of 200 load shapes, we use the clustered labels (i.e. shape indices) to represent each household's load shape pattern. Specifically, we calculate the frequency of the load shapes for a household and represent them as a 200-dimensional vector. For example, during a calendar year, if the home i repeated "shape 1" for 23 days, "shape 2" for 17 days, "shape 200" for 325 days, then we have the vector $r_i = [23, 17, \ldots, 325]$ to describe the one-year pattern of home i, where $r_i \in \mathbb{R}_+^{200}$. Referring to Figure 2.1b, we stack all households' load patterns into an n-by-200 matrix M_r where n is the number of homes.

We apply the LDA method to extract a few distinct and representative attributes of load shapes. To determine how many attributes are appropriate to both capture all consumption patterns while also being sufficiently representative, we prescribed 10 attributes and then merge the neighboring attributes together using a bottom-up approach, i.e. by calculating correlations of attributes and projecting them down to lower dimensions. After consolidating similar attributes, this ultimately yields six representative attributes that are quantitatively and descriptively distinct in terms of daily consumption patterns (Figure 2.2). More details including why we chose 6 attributes are covered in section 2.5.3. We observe that *attribute 0* has the peak consumption around 10pm-11pm with low energy use during the day. A similar pattern is also observed in *attribute 2* but with a longer time span of late-night consumption, whereas *attribute 1* has a peak consumption around 6pm-7pm with lowest energy use around 2am-3am. Distinct from other attributes, *attribute 3* has the highest energy usage in the afternoon from 12pm-5pm and *attribute 4* displays peak usage around 7am-8am in the morning. Finally, we find *attribute 5* has comparatively low variation in daily usage.

With these six summarized attributes, each home is then characterized by assigning a 6-dimensional vector where the value at each entry represents the corresponding attribute weight. The attribute weight at the k-th entry indicates how likely a home possesses *attribute-k*. In our experiment, many households share similar attribute weights over a year, so we therefore define lifestyles using k-means clustering to obtain a stable result of six lifestyles. We found that six lifestyles were sufficient to cover

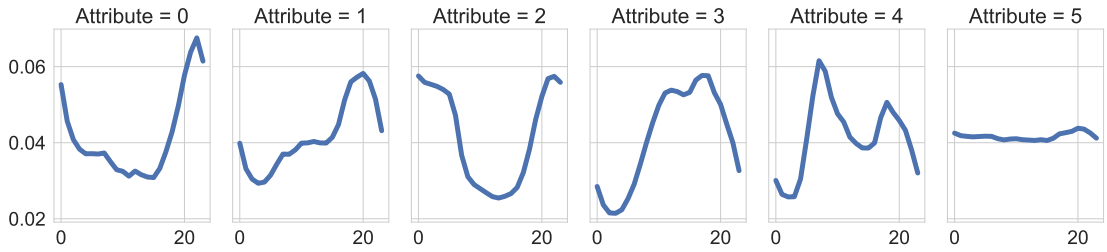


Figure 2.2: Attribute shapes. Each attribute shape is a weighted sum of 200 dictionary shapes, where the weights are the normalized probabilities of each shape's occurrence.

the heterogeneity of daily lifestyle patterns (shown in Figure 2.16 and Figure 2.17). In Figure 2.3, we plot each lifestyle as a black line that represents a weighted average of different attribute weights depicted by the thickness of the dashed lines. Given their load shape characteristics, for ease of reference we assigned names to each of the lifestyles from left to right as *active mornings, night owl, everyday is a new day, home early, home for dinner*, and *steady going*. In naming these lifestyles, we use the following as descriptive justification: *active morning* has a distinguishing characteristic of energy use in the morning time period; *night owl* has energy use in the very late night and very early morning with little morning through evening usage across days; *everyday is a new day* displays substantial heterogeneity in daily energy use patterns across different days; *home early* is distinguished by its late afternoon use; *home for dinner* has energy use concentrated in the evening; and the *steady going* lifestyle has use that remains relatively stable throughout the day. We have no additional, non electricity-use information about these households to verify or justify these lifestyle names, a challenge confronted by other "unsupervised" learning applications [39, 6].

The *home for dinner* lifestyle is the most frequently occurring lifestyle among our sampled residential households, accounting for 39.7% of the households in our dataset. The next most frequently occurring are the *home early* and the *everyday is a new day* lifestyles, which account for 19.1% and 19.9% respectively, followed by *active mornings* and *night owl*, both of which account for approximately 8% of the sample. At 4.7%, the least frequently occurring lifestyle is *steady going*. Although the mean representations of *everyday is a new day* and *home for dinner* are similar,

they are different in that the mixture weights of the attributes are evenly distributed for *everyday is a new day* but the weights for *home for dinner* are concentrated on *attribute 1* (Appendix Figure 2.17).

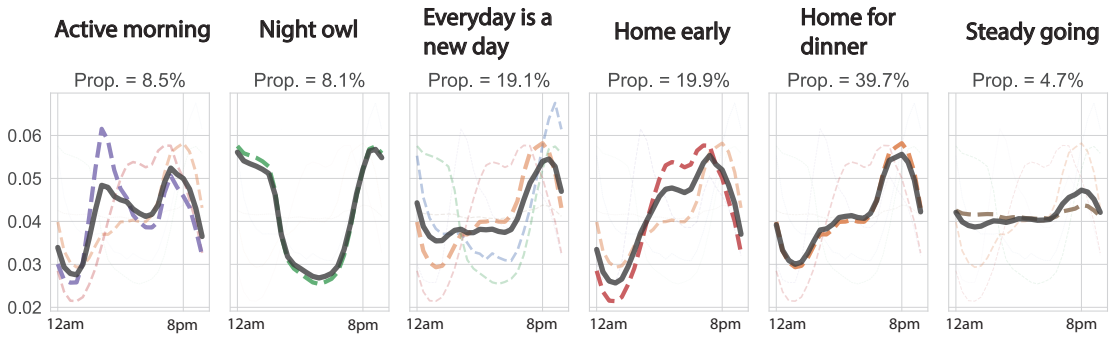


Figure 2.3: Lifestyles. From left to right, they are *active morning*, *night owl*, *everyday is a new day*, *home early*, *home for dinner*, and *steady going*. The different thickness of the dashed lines indicate different composition weights for corresponding attributes.

While different households have different lifestyles, we also observe that a single household can also display multiple lifestyles over the course of a year. For example, one set of lifestyle patterns could be related to the presence of children in the home, such as when children are on break during the summer months and in school during the fall. The change of lifestyles of a household relates to its energy use that could be associated with a household members' behaviors (e.g., occupancy) under different time horizons (e.g., months, seasons, years, etc.). To this end, we next examine how these energy use behaviors change across time by choosing season as a convenient unit of measurement, as we only have access to one year of data and cannot observe changes over longer time periods. Therefore, we partition our one year's worth of data into four seasons: Autumn (Sept. - Nov.), Winter (Dec. - Feb.), Spring (Mar. - May), and Summer (June - Aug.) and run lifestyle analysis for seasonal time intervals. The lifestyle transitions of households across seasons is displayed in Figure 2.4.

We find that the *home for dinner* comprises a larger proportion of household across seasons compared to the other lifestyles. Such a seasonal phenomenon also matches the previous findings in the observations across the entire year (in Figure 2.3). Each season contains households with all six lifestyles except for summer which does not

contain any households with the *steady going* lifestyle. One reason could be that the relatively flatter usage profile of *steady going* is particularly uncommon during summer months because thermal comfort-related consumption—such as HVAC usage—tend to be turned on and off for a multiple hours across a day, yielding a more volatile daily load shape. Whereas in the winter, many homes rely on gas–heating and regulation of thermal comfort could yield a flatter pattern of electricity use. A detailed description of household sample membership in lifestyles across the four seasons is provided in Appendix Table 2.4. Furthermore, we find that some households stay in a single lifestyle across all seasons, whereas other homes switch between two or more lifestyles over the course of four seasons. Such an observation motivates us to investigate the distinctions between those lifestyle-consistent households and lifestyle-changing households.

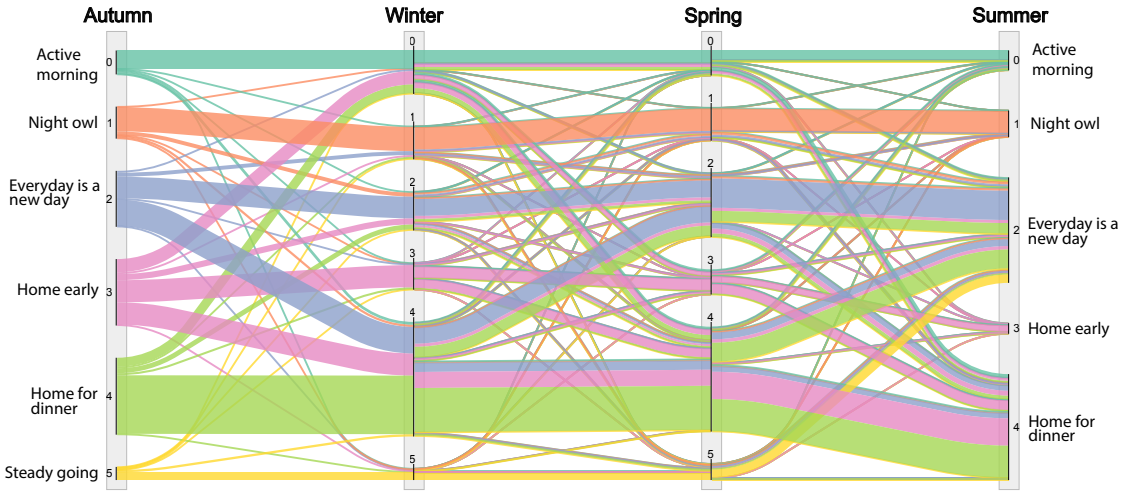


Figure 2.4: Seasonal transition of lifestyles spanning from Autumn 2010 through Summer 2011. The color of the lines represent the lifestyle designation in Autumn and tracks groups of households across time. The thickness of the lines represent the proportion of total households in each lifestyle at each seasonal interval, with wider lines indicating more households and thinner lines fewer households.

2.3.4 Energy lifestyle analyses

To understand what determines different lifestyles, we explore a number of energy use characteristics derived from raw smart meter data from households. Unlike many other studies [31, 6, 63], our energy use characteristics are generated using raw energy data from households. These energy use characteristics (also known as features) are directly derived from raw smart meter data, which does not rely on the previously generated load shape dictionary or energy attributes. We first illustrate the features associated with corresponding lifestyles summarized across one year. Next, we explore the changes overtime of various features of lifestyles across seasons, and develop a way to identify those households that change lifestyles across seasons (*Changer*) and those who do not (*No Changer*).

Features of energy use

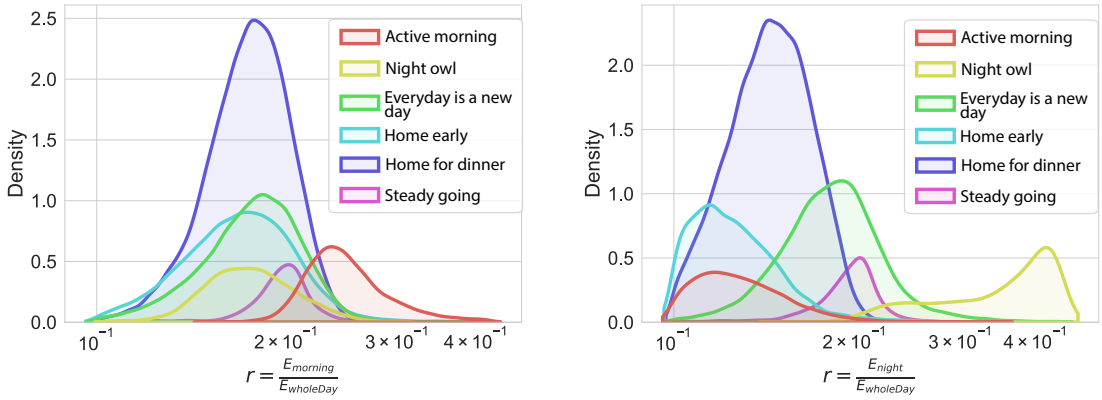
Once we have constructed our lifestyles, each home is automatically associated with a single lifestyle that matches its energy use pattern. Conditioning on these lifestyles, we consider all households that are labeled in the same lifestyle as a group, yielding six groups among all households in our sample. We find that each group of households has unique distributions of certain energy use features. These features are extracted from raw electricity use from households, such as mean daily energy use, ratio of morning to whole day energy use, and peak hour frequency (normalized), described in detail in Table 2.2. We are interested in these distributions of features, especially when they have distinct characteristics, since they can be used to identify different lifestyles from the pattern of features instead of observing a whole year of household consumption.

We present a few examples showing that certain lifestyles can be distinguished from feature distributions (e.g. Figure 2.5a and Figure 2.5b). Specifically, Figure 2.5a indicates that the *active morning* group has the highest ratio of morning (6am - 10am) to whole day energy use where the mean is approximately 0.24 and with a small portion of the homes having a ratio value over 0.4. The *home for dinner* group has a mean ratio value of approximately 0.18 with a substantial portion of homes having an even lower value of 0.1. Other lifestyles such as *night owl*, *everyday is a new*

Table 2.2: Description of features of electricity use

Feature	Description
E_{day}	mean of daily energy use
E_{hour}	mean of hourly energy use
E_{peak}	mean energy use of peak hour in a day, equivalent to E_{max}
E_{base}	mean base energy use of a day
E_{min}	mean of min energy use of a day
$E_{morning}$	morning energy use between 6am to 10am
E_{noon}	morning energy use between 10am to 2pm
$E_{evening}$	evening energy use between 6pm to 10pm
E_{night}	night energy use between 10pm to 2am
$E_{wholeday}$	24 hour energy use
r_{base}	base load ratio, i.e. mean of $\frac{E_{base}}{E_{day}}$
$r_{min2max}$	mean ratio of min hourly load divided by max hourly load, i.e. mean of $\frac{E_{min}}{E_{max}}$
r_{m2w}	mean of morning energy use divided by whole day energy use, i.e. mean of $\frac{E_{morning}}{E_{wholeday}}$
r_{n2w}	mean of noon energy use divided by whole day energy use, i.e. mean of $\frac{E_{noon}}{E_{wholeday}}$
r_{e2w}	mean of evening energy use divided by whole day energy use, i.e. mean of $\frac{E_{evening}}{E_{wholeday}}$
r_{ni2w}	mean of night energy use divided by whole day energy use, i.e. mean of $\frac{E_{night}}{E_{wholeday}}$
π_j	multinomial distribution over 24 hours showing the normalized frequency of peak hour occurrence. The j takes value from 0, 1, ..., 23, indicating j-th peak hour in a day

day, and *home early* have the mean ratio value between 0.16 - 0.19. In short, these distributions are consistent with initial insights about our lifestyles as *active morning* has higher energy use than other lifestyles in the morning time period. The *night owl* style has the highest ratio of night (10pm-2am) to whole day energy use where the mean is approximately 0.44, suggesting that many homes use energy between 10pm-2am, accounting for approximately 44% of the whole day use in that lifestyle group. Other styles have mean ratio values below 0.15 except for *everyday is a new day* and *steady going* lifestyles, both of which have either non-trivial energy consumption during the night period because of switching between different energy attributes or have a flatter shape associated with the energy attributes.

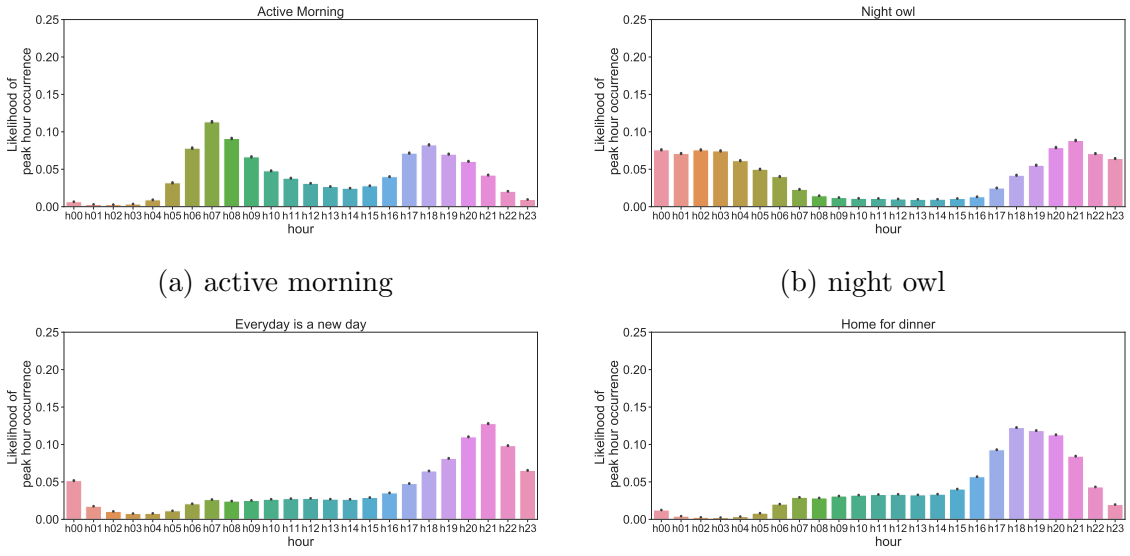


(a) ratio of morning to whole day energy use (b) ratio of night to whole day energy use

Figure 2.5: Load features of different lifestyles. (a) suggests that *active morning* style has a higher ratio of morning to whole day energy than other lifestyles. (b) reflects that *night owl* has a significantly higher ratio of night to whole day energy than other lifestyles. One potential benefit of looking into these ratios is that we can quickly classify some homes into corresponding lifestyles without observing their annual consumption data.

Apart from the intra-day's ratio of energy use, we compare the peak hour occurrence of the different lifestyles. We present four typical lifestyles (Figure 2.6) because they are representative and prevalent among households. Figure 2.6 suggests that the distributions of peak hour frequencies align with lifestyles even though the frequency of occurrences are extracted from raw energy use. For example, the pattern of peak hours frequency for *night owl* (Figure 2.6b) closely matches with its lifestyle curve

(Figure 2.3). Similar matches can also be found in other lifestyles like *active morning*, *everyday is a new day*, and *home for dinner*. Such descriptive cross-validation in peak hours demonstrates the value of our lifestyle framework, while also building an understanding around inductive features for various lifestyles. We present additional summaries of features in 2.5.5.



(a) active morning

(b) night owl

(c) everyday is a new day

(d) home for dinner

Figure 2.6: Peak hour distributions. Each sub-figure shows the averaged frequency of peak hour occurrence for all homes in the corresponding style group. We illustrate four lifestyles because these lifestyles are prevalent among the households.

As the distributions of features differ substantially among various lifestyles, we expect that lifestyles can be identified by using load-related features. To assess this, we establish a classification problem where the lifestyle of household i is the label y_i and the features are the observed predictors $\boldsymbol{x}_i$. We therefore learn a mapping f such that $y_i = f(\boldsymbol{x}_i), \forall i \in 1 \dots N$ where N is the number of samples. For interpretability and robustness, we apply random forest (RF) as our classification model. After splitting the training, validation, and test sets using the portions of 70%, 10%, and 20%, followed by selecting and calibrating features, we then fit a RF model with a classification accuracy of 68.5% on average (in Figure 2.7a). We find that *night owl* is the easiest lifestyle to classify having approximately 82% accuracy. In contrast, *home*

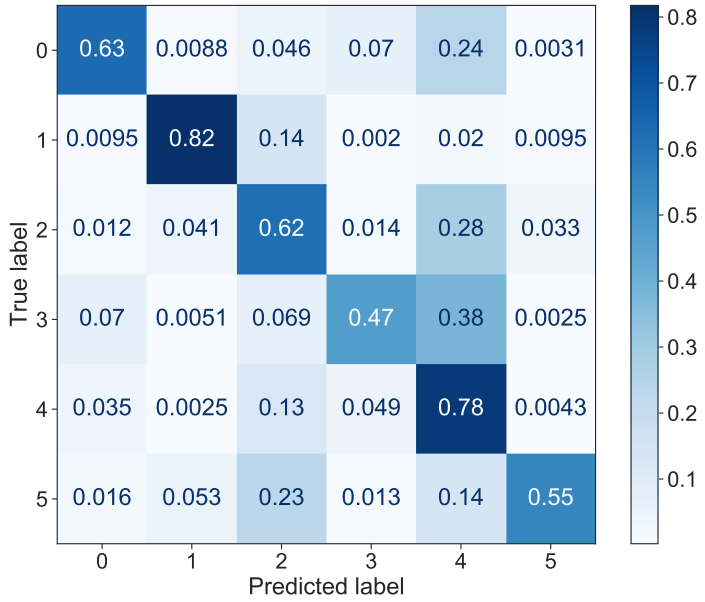
early is the most difficult lifestyle to model with 47% accuracy since a significant portion is miss-classified as *home for dinner*. These observations are also supported by the classification results of precision, recall, and F1 score (shown in Table 2.5).

In addition to comparing the feature distributions of lifestyles and classifying each lifestyle based on household energy consumption, we investigate what features have important roles in determining lifestyles. We use a model-agnostic permutation importance score described in [64, 65] to estimate the importance of the features in our random forest model, and discover that the features constructed as various ratios play major roles in identifying lifestyles (Figure 2.7b).

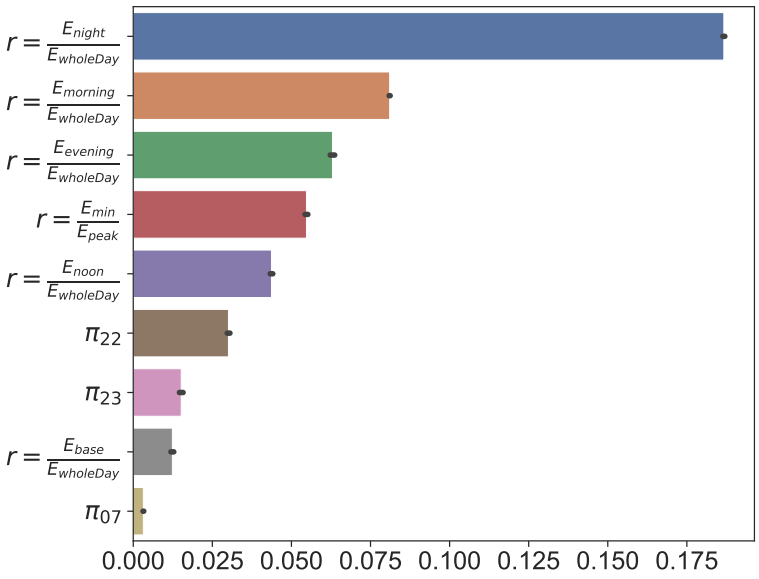
We find the mean ratio of night to whole day usage is the most important feature, contributing to approximately 18% of additional accuracy compared to a case where the ratio is identically distributed (i.e. random assignment), followed by the mean ratio of morning to whole day that contributes an additional 8% accuracy. We also observe that the peak hour frequencies at 7th, 22nd, 23rd hour are non-trivial in determining the lifestyle, suggesting that the peak consumption in the night around 10pm-11pm and in the morning around 7am are important features. As an additional robustness step, we verify that these top features are not highly correlated (see Figure 2.27). We further verify these results by running multinomial logistic regression models and find statistical significance for these ratio features. We provide more details in section 2.5.7.

Dynamics in energy lifestyles across time

In addition to these year-specific patterns, we also compare the distributions of features at the seasonal-level because certain households may change lifestyles (i.e., Changer) or may not change lifestyles (i.e., No Changer) across a single year period. Since the *steady going* lifestyle does not occur in the summer (Figure 2.4), the following analysis is focused on the remaining lifestyles (Figure 2.3). First, we assess the characteristics of No Changer households in terms of load-related features. In particular, we compare both the ratio of morning to whole day usage and the ratio of evening to whole day usage across four seasons to check the stability of the feature distribution among various lifestyle groups. We find that *active morning*, *everyday*



(a) Confusion matrix of classifying lifestyles. The lifestyle label 0 to 5 are *active morning*, *night owl*, *everyday is a new day*, *home early*, *home for dinner*, *steady going*.



(b) Feature importance

Figure 2.7: Classification results. (a) The confusion matrix suggests that *night owl* has the highest accuracy at 0.82. In contrast, the *home early* has the lowest accuracy, 0.47, since a majority of samples are misclassified as *home for dinner*. (b) indicates the top nine important features needed to correctly classify a home's lifestyle. In this case, the ratio of night to whole day energy use is the most important feature.

is a new day, *home early*, and *home for dinner* have very stable distributions across four seasons. Consistent with the lifestyle name, *active morning* is influenced by the morning to whole day ratio value (approximately 0.26) compared with any other lifestyle's mean ratio (that is below 0.2) (Figure 2.8). Although *night owl* households tend to keep this lifestyle across multiple seasons, we note that the ratio of morning to whole day usage of the *night owl* lifestyle shifts toward smaller values in the summer compared to other seasons, indicating some homes either increase their whole day energy use or reduce their consumption in the morning period during the summer. Such a pattern matches with previous discoveries [66] that large consumption or late morning activities are more likely to occur in summer for residential households. To confirm the No Changers' stability of load characteristics, we further compared the ratio of evening to whole day usage across the seasons in Figure 2.9. We observe that all lifestyles have stable distributions of this ratio, with means located between 0.27 to 0.32, except for the *night owl* style that has a mean of 0.19 in summer and 0.34 in winter. Some homes in *night owl* lifestyle indeed show larger consumption after midnight in the summer, whereas in the winter the night usage pattern begins earlier in the night. Other features, such as mean load and peak load, also demonstrate the stability of No Changer households in various lifestyles (in Figure 2.24 and Figure 2.23).

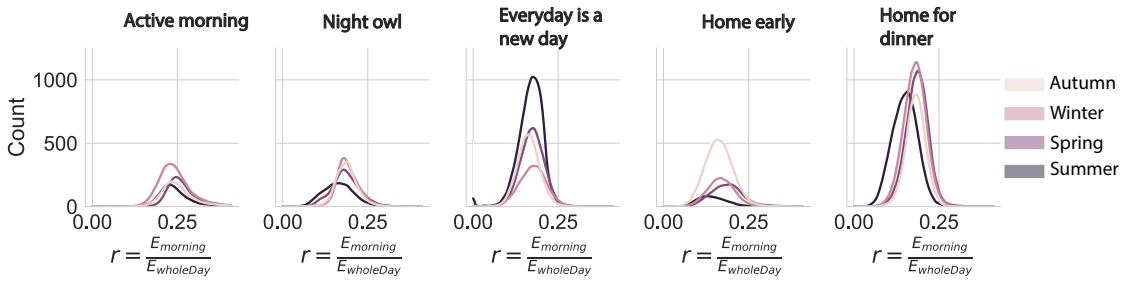


Figure 2.8: Ratio of morning to whole day of energy. The distribution mode of this feature is relatively stable for No Changer, except during summer season a small portion of population shifts the ratio a little in different lifestyles.

Second, we compare the distributions of load features between Changer and No Changer to understand the difference between these two groups across various lifestyles.

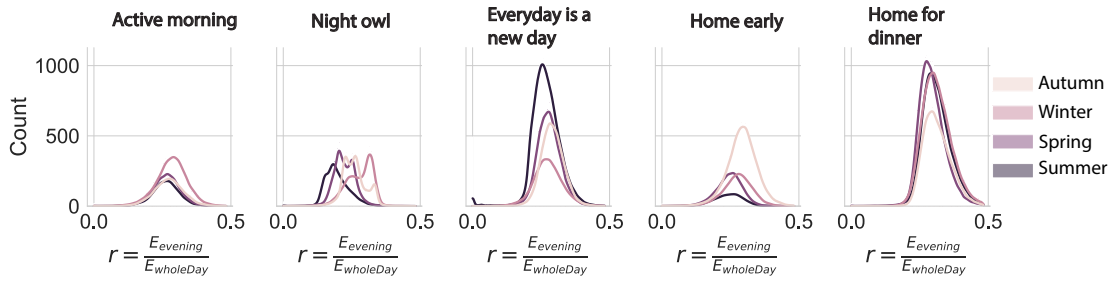


Figure 2.9: Ratio of evening to whole day of energy. The distribution mode of this feature is relatively stable for No Changer in many lifestyles. Yet in the Night owl lifestyle, some homes still change the usage across four seasons indicated by the second panel above.

Specifically, we evaluate these two groups given a lifestyle and a season, and then expand the evaluation over multiple seasons and lifestyles. For example, the distribution of the ratio of morning to whole day usage is expressed in Figure 2.10, which suggests three insights. First, in the *active morning* lifestyle, the Changers' mean is lower than the No Changers' mean over four seasons. Such a pattern indicates that No Changers tend to consume more in the morning than Changers. Second, overall the No Changers have lower means than Changers for the *night owl*, *everyday is a new day*, *home early*, and *home for dinner* lifestyles in four seasons. When comparing the composition of these attributes, No Changers in those lifestyles have higher consumption in the afternoon than in the morning, indicating that morning usage is relatively small. Thus, the Changers could have higher morning usage because they are not restricted to a single lifestyle. Third, the population of Changers is larger than that of No Changers. Many Changers switch their styles between *everyday is a new day*, *home early*, and *home for dinner*. In the winter, Changers are mainly concentrated in the style of *active morning* and *home for dinner*. In contrast, in the summer, Changers are mainly located in *everyday is a new day* and *home for dinner*. Alternative comparisons using the base–to–peak ratio (in Figure 2.25) also suggest that No Changers differ from Changers across seasons.

Considering these distinguishing distributions of characteristics between *Changers* and *No Changers*, we then classify these two groups in the context of different lifestyles, because being able to distinguish whether a household is a Changer or No

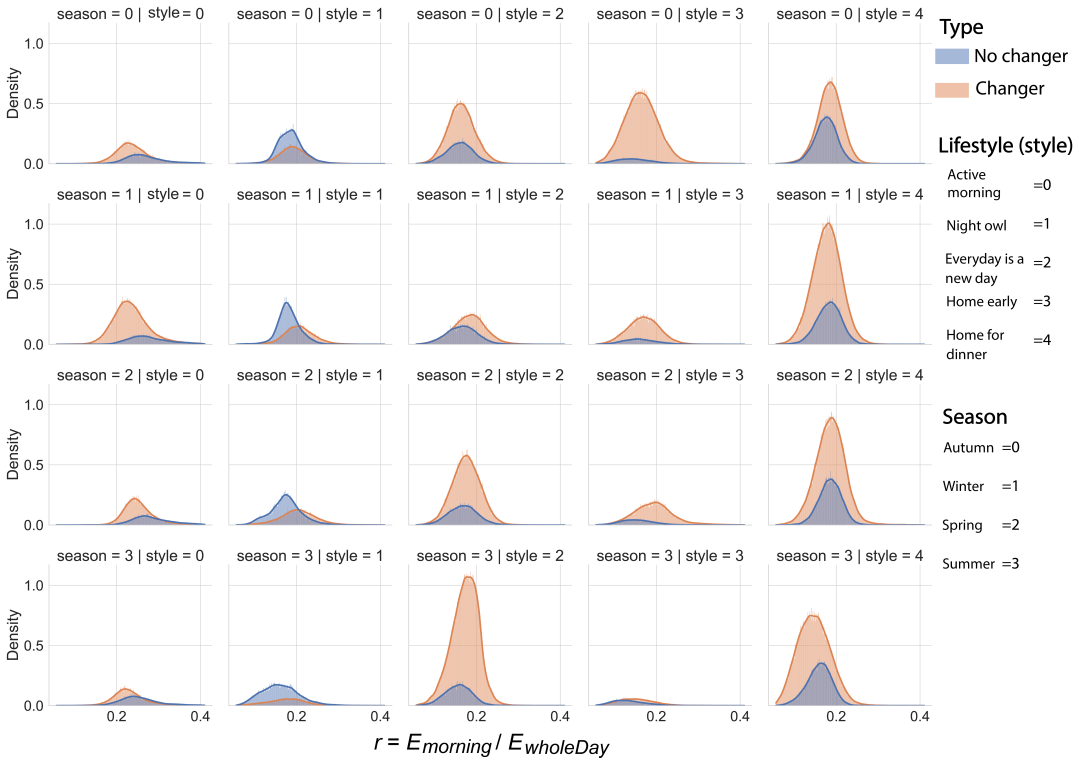


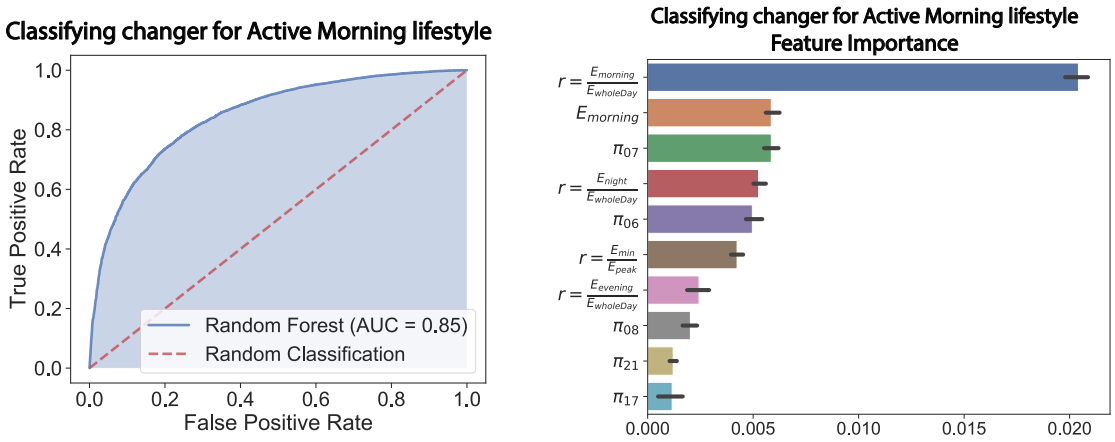
Figure 2.10: Ratio of morning to whole day of energy. Lifestyle is abbreviated to "style" for visualization purposes.

Changer can help energy service providers to identify different types of households and provide customized services (e.g., different energy saving rebates). To accurately classify a Changer or No Changer in each lifestyle, we label the No Changer homes as 0 and label the Changer homes (who possessed the corresponding lifestyle once and then switched to other styles) as 1, and then apply a random forest model to this binary classification problem. For example, the style of *active morning* achieves 87.9% identification accuracy (Table 2.6). Because the positive and negative samples are not evenly distributed, the binary decision can be adjusted for a low false positive rate, as shown in the receiver operating characteristic (ROC) curve in Figure 2.11a (where the red dotted line denotes performance of random selection). The area under the curve (AUC) is 0.85, meaning that a randomly selected positive example (i.e. a home of a Changer) is more likely to be a Changer than a randomly selected negative example (i.e. a home of a No Changer) with probability 0.85.

Once the classifier is fitted to identify a Changer, we evaluate the top determinant features by again using the permutation importance method. Figure 2.11b suggests that the ratio of morning to whole day usage, the morning energy use, and the peak hour frequency at the 7-th hour (7am-8am) are among the top three most important features. Such findings indicate that the pattern of energy consumption in the morning period can largely determine whether a household is a Changer or No Changer in the active morning lifestyle. We also verify that these top importance features are not highly correlated (Figure 2.27), which demonstrates the robustness of our results regarding important features.

We assess both the classification performance and feature importance when identifying Changers and No Changers in other lifestyles (Figure 2.28, Figure 2.29). The results show that classifying Changers in the *night owl* lifestyle has the highest AUC value of 0.97, and doing so in the *home for dinner* lifestyle has the lowest AUC value of 0.77. Such different performances of the AUC metric suggest that identifying Changers in the *night owl* group is much easier than identifying Changers in the *home for dinner* group (Figure 2.28a and Figure 2.28d). For feature importance, we find individual lifestyles to have their own prominent features that determine Changers separately, but that the features that are characterized by various ratios of energy

use play important roles in all lifestyles (Figure 2.29). In general, features related to certain time spans within a day (such as ratio of evening to wholeday energy use) can be applied to identify whether a household is a Changer or not, and have a better performance compared to volume-based features (e.g. base load and hourly mean load, etc.).



(a) By varying the classification threshold, we can trade off between true positive rate (TPR) and false positive rate (FPR). The receiver operating characteristic (ROC) curve shows the TPR and the FPR are significantly higher than random classification, having an AUC of 0.85.

(b) The top 15 features are first selected according to the F-value from χ^2 tests between labels (changer or no changer) and features. Then top 10 important features are listed after we permuted the features and fed them into the fitted random forest model. In *active morning* style, the ratio of morning to whole day is the most important feature.

Figure 2.11: Identifying Changer v.s. No Changer

2.4 Discussion and Conclusion

2.4.1 Targeting and Tailoring customers

In this research, we present a new approach for constructing dynamic energy lifestyles by applying LDA to residential electricity demand data. Our framework is highly scalable and extensible, while also being flexible enough to accommodate different time intervals and completely new sources of residential energy data from other locations

and contexts. Using this dynamic lifestyle approach, we can greatly simplify the interpretation of energy lifestyle patterns by using a method that generates a sparse number of energy attributes that are then used to generate a manageable set of energy lifestyle profiles. We show this process of generating energy lifestyles is robust to multiple load shape dictionary inputs and time intervals. We also demonstrate that these derived energy lifestyles can be associated with certain energy use characteristics, even though these energy use characteristics were not originally applied in constructing the lifestyles themselves.

We use these energy use characteristics to further interpret these lifestyles and provide insight into how such an approach toward lifestyle analysis could be used in practice. This energy lifestyle analysis approach can also be applied across different time horizons, allowing for applications at varying time intervals to examine temporal dynamics. While in our experiment we applied a seasonal time interval, shorter (e.g., monthly) or longer (e.g., yearly) time horizon energy lifestyles can also be estimated—dependent on data availability. Such an approach provides the ability to generate meaningful insights that can be applied to a wide variety of energy program designs and use cases. This approach may be particularly useful for entities such as utilities who need to under stand household energy lifestyles and lifestyle change patterns across time. One example is exploring the demand flexibility for households. Such a demand flexibility is not only limited to considering electricity use timing throughout a day such as TOU programs [25, 26], but also reveals the day–to–day or even seasonal variations of energy use for households.

2.4.2 Potential applications of the dynamic lifestyles framework

We have identified three potential applications for this lifestyle analysis approach, each considering a different aspect of energy program design. The first application is in identifying households with lifestyle patterns that are most appropriate for installation of behind–the–meter resources, such as residential solar and battery storage systems. Taking the example of households with the energy lifestyle *home early*, these households may be particularly well–suited for rooftop residential solar as they have

a pattern of usage that begins in midday, when solar energy potential is higher. A household where usage tends to peak later in the day and during evening hours with less solar energy potential, such as *home for dinner*, would be less suitable for targeting residential solar unless it was combined with battery storage (i.e., solar plus storage system) [67].

For demand response programs, certain energy lifestyles we derived from our experimental data suggest differing demand flexibility for households, especially when considering demand responsiveness to time-of-use pricing, which typically occurs during weekdays when system-level demand is highest, such as in the late afternoon and early evening. For households in *everyday is a new day*, their daily energy use is highly varied. This suggests that these households could be more able to change their daily energy use patterns, making them flexible in their demand because their energy use patterns are less structured compared to other energy lifestyles, such as *home early* and *home for dinner*. Energy lifestyles that are less flexible, such as *home early* and *home for dinner*, however, may be better suited for energy efficiency programs, because their lifestyle energy use patterns indicate stable electricity patterns with little day–to–day variation.

While both of these examples are related to households that display relatively static energy lifestyle patterns, how these patterns differ across time is also important for potential applications in practice. First, if a household always displays a particular energy lifestyle pattern, this suggests that the household has a higher affinity toward the pattern of energy use within this lifestyle compared to a household that displays a change in lifestyle across time. Next, the number of lifestyle profile changes that a household undergoes on a seasonal basis, and the variety of these lifestyles, imparts important information about the household. Households that are constantly undergoing change will likely be difficult to target customers with demand response programs [51] given the instability of daily usage patterns. However, such a household may be a better candidate for an energy efficiency program, such as smart thermostat/AC.

While these examples have been targeted to energy provider applications, there are also opportunities to use this energy lifestyle analysis framework to inform energy

intervention design, where households attempt to change their lifestyles to promote energy use patterns that save them money while also lessening their burden on the grid and carbon emissions. To do so, households that have a particular lifestyle with peak demand that corresponds with system demand, such as *home for dinner*, could attempt to change their usage to a different lifestyle pattern, such as *steady going* or *active morning*, with less usage concentrated during peak system demand periods. This energy lifestyle approach could then be used to determine if there is a shift in lifestyles, and also could become the basis in which to assess whether the household had successfully implemented this change. Moreover, such an approach may be used for households to quickly monitor their own energy lifestyle and make adjustments based on changes in the home or other new activity patterns [68]. In this respect, communicating information about energy use to customers via their lifestyle profile may be more impactful than other forms of more traditional energy use informational summaries (e.g., monthly kWh or energy cost).

Given the wide applications of this dynamic lifestyle approach, for which we have only provided a few examples, as well as the ability for iterative updating of energy lifestyles, we see great potential for building and extending this framework. However, our approach has some limitations. First, while we are able to verify these energy lifestyles using other energy use characteristics that were not included in the formulation of the energy lifestyles, we do not have an additional external measure to verify the presence or absence of this lifestyle based on other, non–energy–use information about household characteristics [6], such as number and age of occupants and patterns of activities in the home. Incorporating such data, if available, would be an important addition to this work and would bolster our framework's ability to provide insights about energy lifestyles. Additionally, the data that we applied in experiments to generate these energy lifestyles is from the early 2010s and therefore do not include recent trends in electricity use patterns within households related to smart home appliances, electric vehicles, and behind–the–meter resources [69], because these technologies were not yet widespread during this time period. To the extent the deployment of these technologies impact the formulation of these lifestyles themselves is not known, but we expect that solar, storage, and electric vehicles have

some discernible impact on lifestyles that we do not capture here. At the same time, something like solar and storage could place a household in the *steady going* lifestyle, so while the formulation of the lifestyles may not be dramatically altered using more recent data, it could be that the proportion of households within a lifestyle will change to reflect new behind–the–meter technology adoption.

2.4.3 Conclusion and next steps

We conceptualized and implemented a new approach for understanding energy lifestyles that can simplify interpretations about household energy use, has a high potential for applicability, can be easily scaled up to larger datasets, and can measure changes in energy use across time. There are four immediate directions for future research as an extension to this work. First, this dynamic lifestyle approach can address a cold start problem in identifying patterns of use for new residential customers. Because this lifestyle approach can identify lifestyles using very sparse data inputs, energy providers could recommend energy program enrollment based on lifestyles after only the first few months of meter activation. Second, this dynamic lifestyle approach can be applied to additional residential datasets spanning different time periods and geographies to explore intra- and inter-yearly patterns in lifestyles as well as the influence of context and climate. Third, some steps of our lifestyles approach can incorporate privacy preserving methods, e.g. differential privacy [70, 71] or generative adversarial privacy [13, 72], to alleviate the concerns of revealing sensitive information of an individual household [14, 16], which is an important direction for future work. Lastly, using information about residential electricity data coupled with demographic and household characteristics, our model can further validate and provide new insights about lifestyles by identifying the characteristics related to different lifestyles and their dynamics across time.

2.5 Supplements

2.5.1 Description of datasets

Our sampled households covers eight different climate zones in California shown in Figure 2.12.

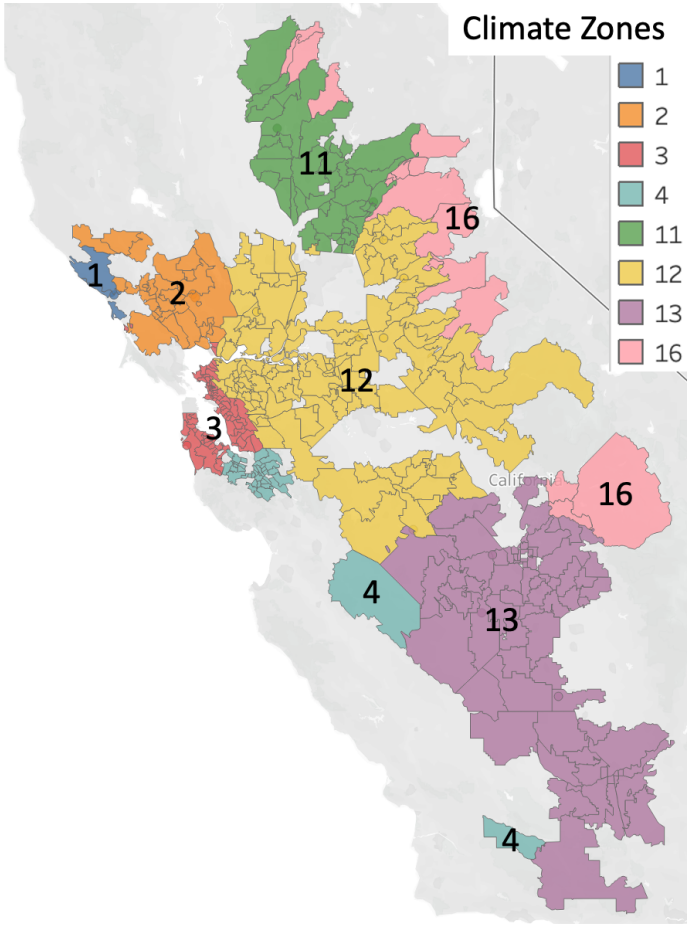


Figure 2.12: Households are located in eight climate zones in California, US.

2.5.2 Details of clustering load shapes

Clustering methods

We apply several clustering methods including k-means, k-medians, hierarchical clustering, and DBSCAN for a thorough evaluation. For the center based methods (e.g. k-means and k-medians), we minimize the following objective (also known as distortion):

$$\min \sum_{i=1}^{N} d\big(\boldsymbol{x}_i, \phi(\boldsymbol{x}_i, C)\big), \tag{2.2}$$

where N is the sample size, d is the distance metric, and $\phi(\boldsymbol{x}_i, C)$ returns the nearest cluster center $c \in C$ to $\boldsymbol{x}_i$. When d is the Euclidean distance and ϕ finds the nearest center using Euclidean distance, we have

$$\min \sum_{i=1}^{N} ||\boldsymbol{x}_i - \phi(\boldsymbol{x}_i, C)||_2^2 = \min \sum_{i=1}^{N} ||\boldsymbol{x}_i - \mu_{c^{(i)}}||_2^2 \quad , \tag{2.3}$$

where $c^{(i)}$ is the cluster label for i-th data point. To express the function ϕ more specifically, the k-means method updates the cluster centers by the following iterations until convergence:

$$c^{(i)} = \arg\min_j ||\boldsymbol{x}_i - \mu_j||_2, \quad \mu_j = \frac{\sum_{i=1}^{N} \mathbf{1}\{c^{(i)} = j\}\boldsymbol{x}_i}{\sum_{i=1}^{N} \mathbf{1}\{c^{(i)} = j\}} \quad . \tag{2.4}$$

The k-medians method differs from the previous k-means clustering when calculating the cluster center. Instead of taking the mean μ_j in equation (2.4), we compute the median as the center $\tilde{\mu}_j$ so that

$$\tilde{\mu}_j = median\big\{\boldsymbol{x}_{i=\{1\ldots N\}}\big\}, \text{if } c^{(i)} = j, \forall i \in 1 \ldots N \quad . \tag{2.5}$$

Hierarchical clustering is an agglomerative (hierarchical) approach, from the bottom individual point to up-level the whole dataset, that builds nested clusters in a successive manner [60, 73]. It has three popular implementations by minimizing different distances (objectives): Ward linkage [74], average linkage [75], and complete

linkage [76]. The Ward's linkage method measures the distance between two clusters, A and B, which is how much the sum of squares will increase when we merge them:

$$\begin{aligned}\Delta(A,B) &= \sum_{i\in A\bigcup B} ||x_i - c_{A\cup B}||^2 - \sum_{i\in A} ||x_i - c_A||^2 - \sum_{i\in B} ||x_i - c_B||^2 \\ &= \frac{n_A n_B}{n_A + n_B} ||c_A - c_B||^2\end{aligned} \tag{2.6}$$

where c_A, c_B are the centers of clusters A and B, and n_A, n_B are the number of points in clusters A and B. Δ denotes the merging cost of putting A and B together. The average linkage calculates the mean distance of all possible pairs of points in two clusters. The complete linkage method calculates the farthest distance of two points allocated in two clusters. In our setting, we pick Ward linkage because it gives a more stable result compared with other two types of linkages.

DBSCAN [61], known as density-based spatial clustering of applications with noise, does not need to specify the number of clusters beforehand. It requires two key parameters, ϵ and $n_{\min}$, which define the neighborhood's distance and the minimum number of points to form a cluster. Higher $n_{\min}$ or lower ϵ indicate higher density to form a cluster. Choosing ϵ and $n_{\min}$ depends on domain knowledge of the data; hence, we evaluate multiple combinations and find it does not scale well for our use case.

Both hierarchical and DBSCAN clustering do not compute cluster centers during iterations; therefore, we add an additional step to calculate a barycenter [77] of the points in each cluster to obtain a representative center. The barycenter is similar to the notion of a center in convex clusters, so we use the sequential averaging technique to compute the cluster center in the context of dealing with time series trajectories [78].

Evaluating different distances

To generate a robust and meaningful dictionary of load shapes, we compare several different distances such as cosine distance, L^1 distance (Manhattan distance), L^2 distance (Euclidean distance), and dynamic time warping (DTW). For simplicity of explanations, we consider two vectors $\boldsymbol{a}$ and $\boldsymbol{b} \in \mathbb{R}_+^m$ (e.g. $m = 24$) in the following

context.

Cosine distance is a measure of similarity between two non-zero vectors of an inner product space. The distance is expressed as

$$d_{cos}(\boldsymbol{a}, \boldsymbol{b}) = 1 - \frac{\boldsymbol{a}^T \boldsymbol{b}}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2} \quad . \tag{2.7}$$

L^1 distance is a measure of the element-wise absolute difference between two vectors. The expression is

$$d_{\ell_1}(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|_1 = \sum_{k=1}^{m} \big|\boldsymbol{a}[k] - \boldsymbol{b}[k]\big|, \tag{2.8}$$

where $\boldsymbol{a}[k]$ is the k-th dimension in vector $\boldsymbol{a}$.

L^2 distance is a measure of element-wise squared gap between two vectors. The expression is

$$d_{euc}(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|_2 = \sqrt{\sum_{k=1}^{m} \big(\boldsymbol{a}[k] - \boldsymbol{b}[k]\big)^2}. \tag{2.9}$$

Dynamic Time Warping distance (DTW) is a method that calculates an optimal match between two given sequences [79]. We adopt a popular implementation that is based on dynamic programming:

$$d_{DTW}(\boldsymbol{a}, \boldsymbol{b}) = D(m, m), \text{ when } D(i, j) = \min \begin{cases} D(i-1, j) + \nu(i, j) \\ D(i-1, j-1) + \nu(i, j) \\ D(i, j-1) + \nu(i, j) \end{cases} , 1 \leq i, j \leq m \tag{2.10}$$

where D is a matrix that records the optimal warping value between the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$; the $\nu(i, j)$ computes the cost between $\boldsymbol{a}[i]$ and $\boldsymbol{b}[j]$ (e.g. the cost is Euclidean distance in this example); and the base case is $D(0, 0) = (\boldsymbol{a}[0] - \boldsymbol{b}[0])^2$.

Hybrid distance: we additionally apply a mixture of the L^2 and DTW distances

to compute the distance between two vectors:

$$d_{hybrid}(\boldsymbol{a}, \boldsymbol{b}) = \gamma d_{euc}(\boldsymbol{a}, \boldsymbol{b}) + (1 - \gamma) d_{DTW}(\boldsymbol{a}, \boldsymbol{b}), \tag{2.11}$$

where the $\gamma \in [0, 1]$ is the parameter to weigh the trade-off between two the distance metrics.

Evaluating clustering performances

To compare multiple clustering methods with different distances, we mainly use two evaluation metrics: *Calinski-Harabaz Index* [62] and *Davies-Bouldin Index* [80]. Both metrics are widely adopted to evaluate clustering models. A higher *Calinski-Harabasz Index* (CHI) relates to a model with better defined clusters, whereas a lower *Davies-Bouldin Index* (DBI) is suggested for a model with a better separation between the clusters. To compare different clustering methods with various distances, we randomly draw 1000 data samples and record the cluster labels that yields the highest CHI and DBI scores when we search the number of clusters from $\{2, 4, 6, 8, 10, 12, 14, 16\}$. We repeat this exercise five times and present the results of the means of CHI and DBI in Table 2.3.

Determining the dictionary size

Once the k-median method with the d_{hybrid} is chosen, we explore the appropriate size of the load shape dictionary. In particular, we tested the size of 100, 200, 300, 400, and 500 load shapes. Such a comparison involves two stages of clustering processes: 1) we randomly partition 60,000 homes into 600 bins where each bin has 100 homes, and then we run clustering on these 100×365 data points for each bin to create 100 cluster centers. 2) Having these 100 clustered load shapes times the 600 bins, we run another round of clustering on 100×600 data points to yield the cluster centers with the size ranging from 100 to 500. Figure 2.13 suggests that a size of 200 reduces the within-cluster distortion dramatically around 20%, which is much more prominent than at other sizes. Thus, we pick 200 clusters as the size of the load shape dictionary.

Table 2.3: Clustering method comparison. We report the means of both *Calinksi-Harabaz Index* (CHI) and *Davies-Bouldin Index* (DBI) after 5 rounds of random tests. A higher CHI indicates a model can yield a better separation of clusters. In contrast, a lower DBI suggests a better separation between the clusters. We see that k-medians with the hybrid distance gives the best clustering performance.

method	distance	$CHI \uparrow$	$DBI \downarrow$
k-means	Euclidean	107.42	4.53
	cosine	102.31	3.87
	ℓ_1	99.51	4.14
	DTW	113.93	3.89
	$d_{hybrid}(\gamma = 0.5)$	116.76	3.67
k-median	Euclidean	109.53	4.50
	cosine	108.11	4.05
	ℓ_1	102.40	4.19
	DTW	115.84	3.82
	$\boldsymbol{d_{hybrid}(\gamma = 0.5)}$	**118.31**	**3.54**
Hierarchical (Ward)	Euclidean	93.21	4.99
	cosine	92.18	4.81
	ℓ_1	90.53	5.16
	DTW	98.65	4.87
	$d_{hybrid}(\gamma = 0.5)$	101.32	4.58
DBSCAN ($\epsilon = 0.1$)	Euclidean	82.44	5.17
	cosine	85.37	5.29
	ℓ_1	80.15	5.18
	DTW	88.03	5.25
	$d_{hybrid}(\gamma = 0.5)$	89.75	5.07

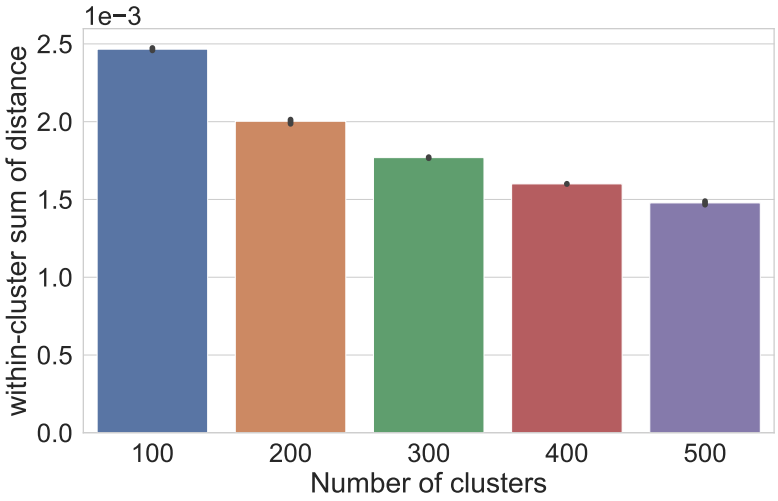


Figure 2.13: Choosing the size of load shape dictionary. When increasing the number of clusters above 200, we have limited marginal gain of reducing the within cluster sum of distances. Thus, we choose 200 as an appropriate dictionary size.

2.5.3 Generating distinct attributes

Before synthesizing the energy lifestyles of households, we need to find the representative attributes that compose the multiple load patterns for households. Thus, teasing out distinct latent attributes of energy usage is a crucial building block. We apply LDA with a prescribed $K = 10$ number of attributes (topics), displayed in Figure 2.14. After fitting the LDA model, we find several attributes are very similar to each other such as *attribute 1* and *attribute 6* in Figure 2.14. A further calculation of the correlation distances between attributes (normalized 24-dimensional vectors) also demonstrates that some attributes are very close and can be merged together (Figure 2.15), where the correlation distances between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with their associated elements means $\mu_{\boldsymbol{a}}$ and $\mu_{\boldsymbol{b}}$ can be expressed as

$$d_{corr} = 1 - \frac{(\boldsymbol{a} - \mu_{\boldsymbol{a}})(\boldsymbol{b} - \mu_{\boldsymbol{b}})}{\|\boldsymbol{a} - \mu_{\boldsymbol{a}}\|_2 \|\boldsymbol{b} - \mu_{\boldsymbol{b}}\|_2}. \tag{2.12}$$

We set the threshold as 0.1 to indicate that two attributes are very similar, and then find the nearest neighbors of the energy attributes based on that criterion.

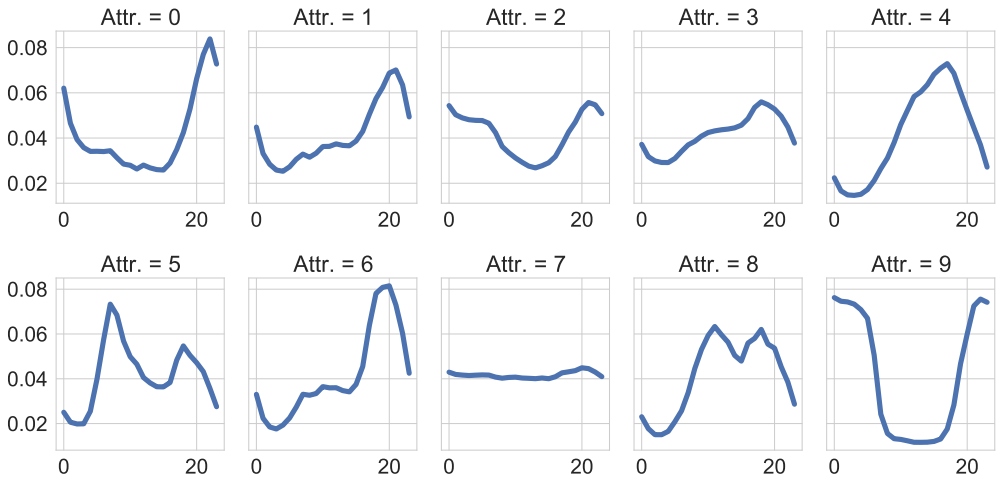


Figure 2.14: Ten attributes are obtained after applying LDA initially. A few center curves are similar, such as Attribute 1 and Attribute 6. We then construct a projection matrix according to correlation distance to reduce the number of attributes (a.k.a, number of topics) down to six.

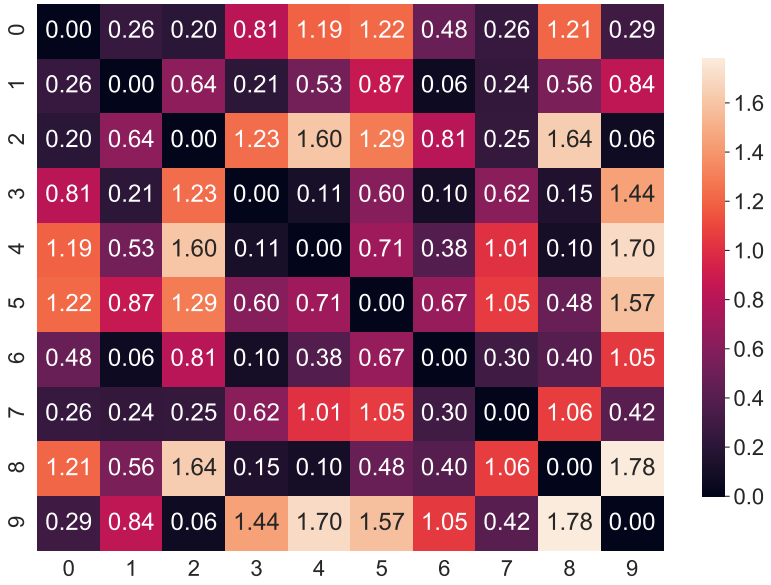


Figure 2.15: Correlation distance heatmap of ten attributes that are obtained from Figure 2.14

Once the neighbors are settled, we merge similar shapes together by 1) constructing a projection matrix A^TA where $A = D_{corr} + I$ and where D_{corr} consists of either zeros or ones, where ones mean when d_{corr} is less than 0.1 in entries, mentioned in equation (2.12) and I is the identity matrix; 2) scanning through columns and pruning the A^TA once the corresponding rows are located. In our experiment, we prune down to six dimensions, because each dimension has its distinct attribute shape (Figure 2.2). Additionally, we qualitatively verify that six attributes are robust for a large population by randomly sampling 2000 homes and comparing their correlation distances on the attribute spaces prior to projection (Figure 2.16). We observe that homes are nested mainly into 5 to 6 diagonal blocks, which supports our previous merge operation of simplifying the energy attributes.

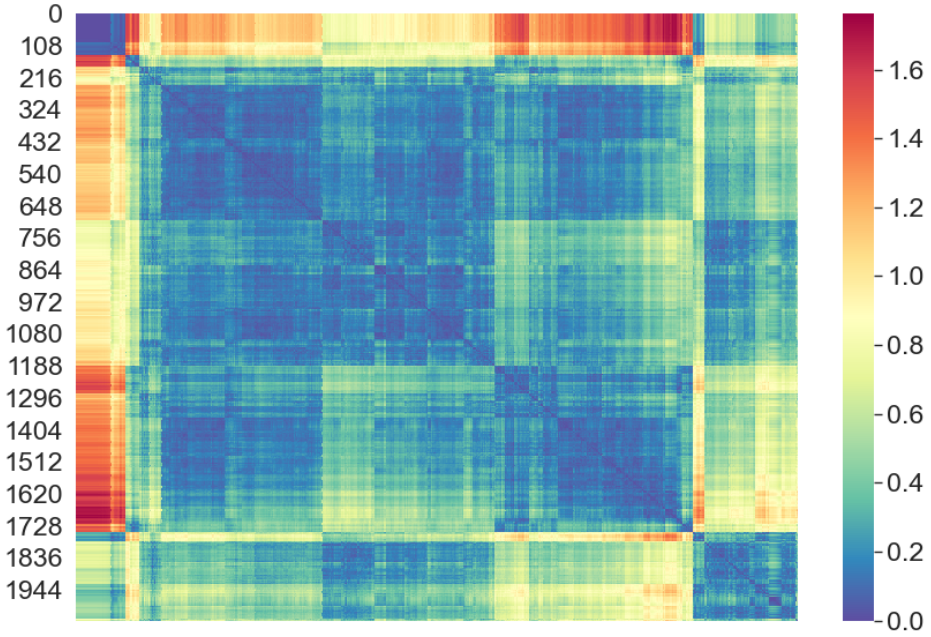


Figure 2.16: Distance heatmap. 2000 homes are randomly sampled and their pairwise correlation distances appear to be segmented into six main blocks along the diagonal.

Having determined the energy attributes, we use the six-dimensional vector to represent each home. In order to obtain prototypical attributes distribution of these homes, we need to segment all the homes using another round of clustering. We use

k-means with $K = 6$, because this setting gives a distinguishable and meaningful result. The corresponding centers of the attribute weights are shown in Figure 2.17.

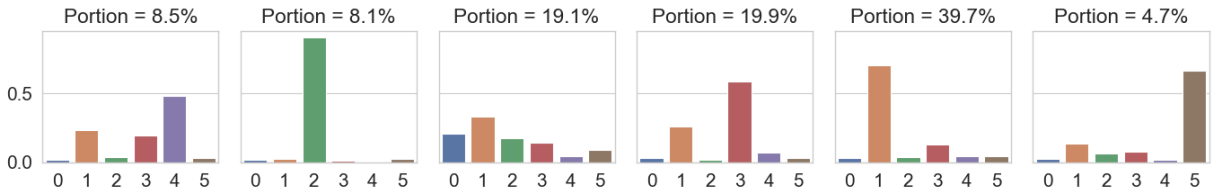


Figure 2.17: Weights of energy attributes

2.5.4 Population change over seasons

We provide detailed population splits of six lifestyles over four seasons in Table 2.4. The numbers are the counts, and percentage values in the parentheses are proportions of the population. From the table, we find that the lifestyle of home for dinner is the most frequently occurring, usually accounting for about 40% of the households except in the autumn when it accounts for 33.02% of the samples.

Table 2.4: Population split of lifestyles in seasons

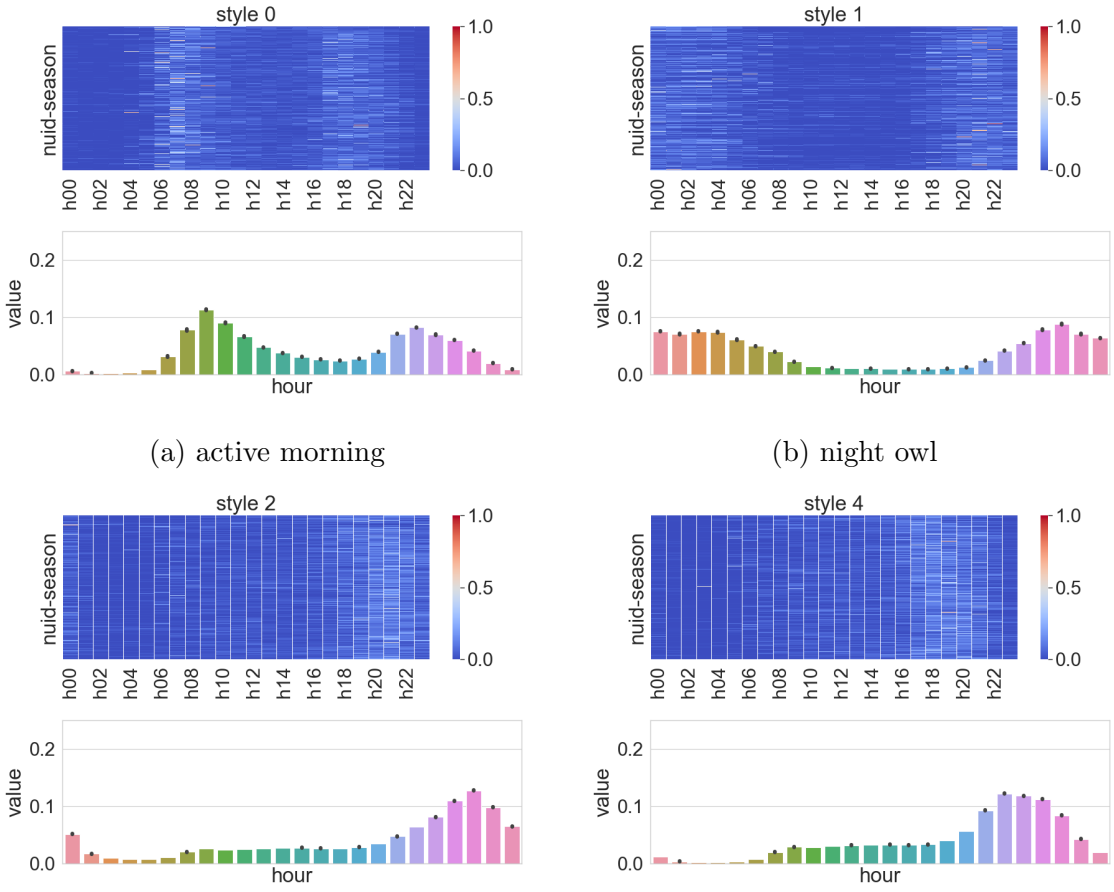
	Autumn N (%)	**Winter** N (%)	**Spring** N (%)	**Summer** N (%)
active morning	4027 (6.71%)	7836 (13.06%)	4713 (7.85%)	3968 (6.61%)
night owl	5174 (8.62%)	4844 (8.07%)	5504 (9.17%)	4557 (7.60%)
everyday is a new day	8632 (14.39%)	5509 (9.18%)	10750 (17.92%)	21311 (35.52%)
home early	18963 (31.61%)	11975 (19.96%)	10254 (17.09%)	4420 (7.37%)
home for dinner	19813 (33.02%)	26084 (43.47%)	24458 (40.76%)	25744 (42.91%)
steady going	3391 (5.65%)	3752 (6.25%)	4221 (7.04%)	0 (0%)†

† We do not observe that households in our samples have a flat pattern of energy use (i.e., steady going lifestyle) across many days in the summer.

2.5.5 Features of energy usage

We show distributions of additional features associated with different lifestyles. The definitions of features are provided in Table 2.2.

First, we provide the peak hour distribution for the *home early* lifestyle in addition to the other styles mentioned in Figure 2.18. Second, multiple year-specific features are displayed in Figure 2.20 over six lifestyles. Finally, we also include additional plots describing the seasonal features. Figure 2.21–2.24 demonstrate the stability of the group of No Changers. These figures cover the different distributions of No Changers across four seasons including morning energy use, evening energy use, peak energy use, and hourly average energy use.



(a) active morning

(b) night owl

(c) everyday is a new day

(d) home for dinner

Figure 2.18: Peak hour distribution over a day (hour 0 – hour 23). In each subfigure, the upper panel shows the heatmap of peak hour frequency when each home in a season is represented by each row stacked by seasons. The lower panel is the averaged frequency of peak hour occurrence for all homes in the corresponding lifestyle group.

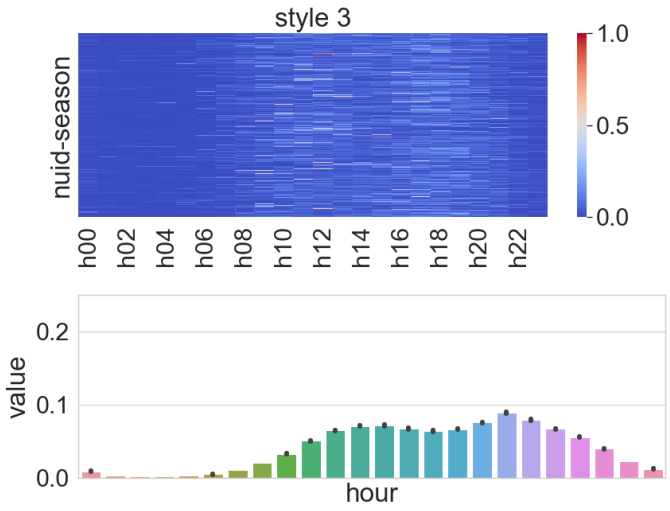


Figure 2.19: Peak hour distributions of *home early* lifestyle.

To provide the detailed comparisons between Changers and No Changers, we show the ratio of night to whole day usage and the ratio of noon to whole day usage in Figure 2.25 and Figure 2.26 because the distributions of those two features significantly reveal the seasonal variations for the Changer group.

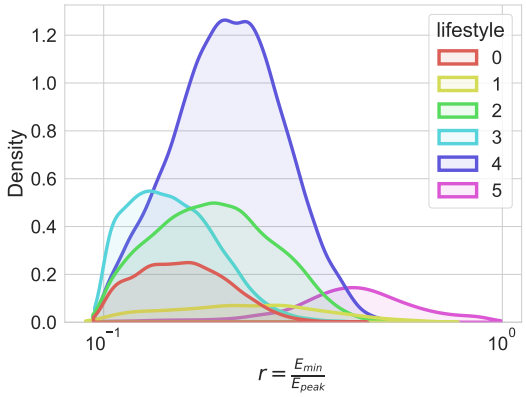
2.5.6 Classification details

Identifying lifestyles

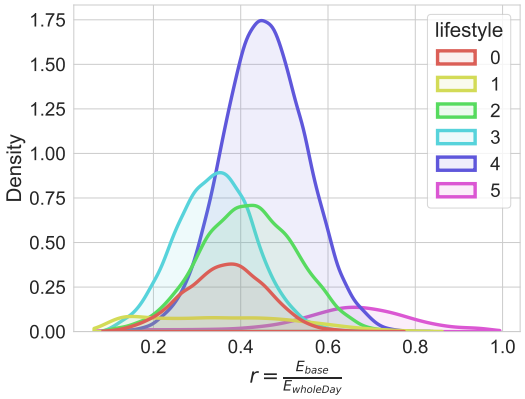
We provide the performance details of classifying lifestyles using random forest fed with load features. The *night owl* has the highest F1 score around 0.84. In contrast, the *home early* lifestyle has the lowest F1 score about 0.55, indicating this is a difficult lifestyle to identify.

Identifying no changer

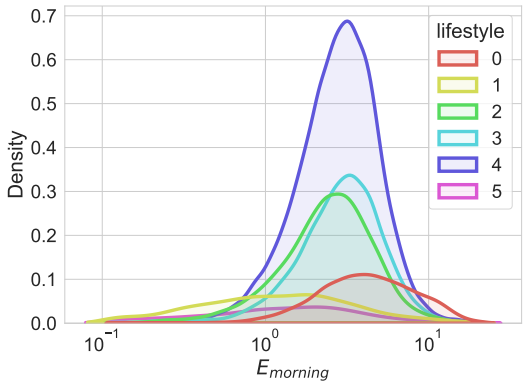
We classify Changer vs. No Changer in each lifestyle. Because the summer season does not have a *steady going* group, we show the other five lifestyles when each of them has a group of No Changers over four seasons (Table 2.6, Table 2.7, Table 2.8, Table 2.9, and Table 2.10). We observe that identifying a Changer is generally easier



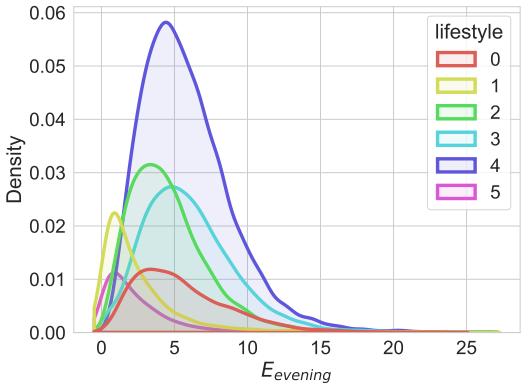
(a) Ratio of hourly min- over max-energy in 24 hours



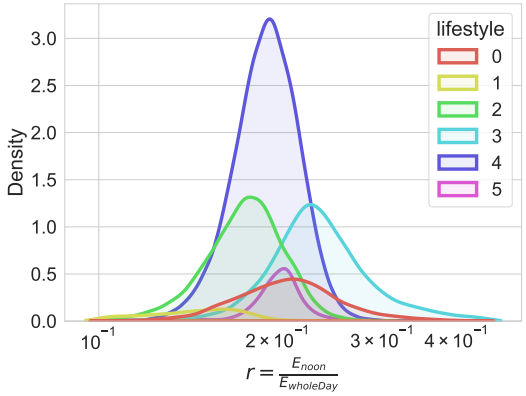
(b) Ratio of base over whole day energy use in 24 hours



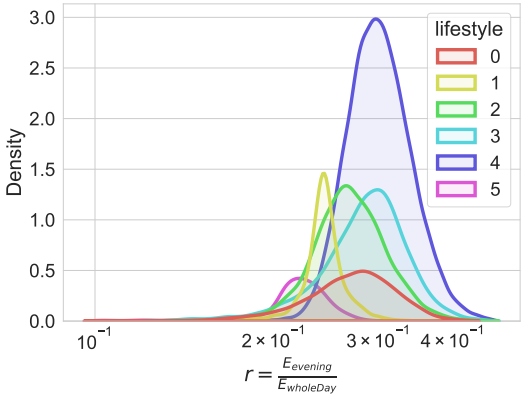
(c) Morning energy use (6am - 10am)



(d) Evening energy use (6pm-10pm)



(e) Ratio of noon over whole day energy use



(f) Ratio of evening over whole day energy use

Figure 2.20: Distributions of different energy usage features characterizing the distinctions between lifestyles.

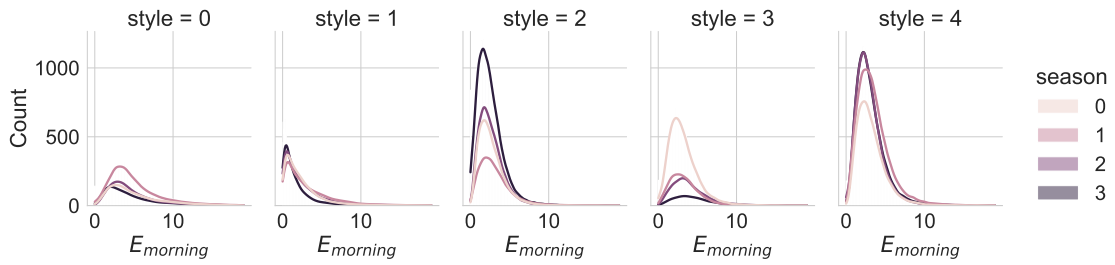


Figure 2.21: Distribution of morning energy use (in KWh) over four seasons for lifestyles

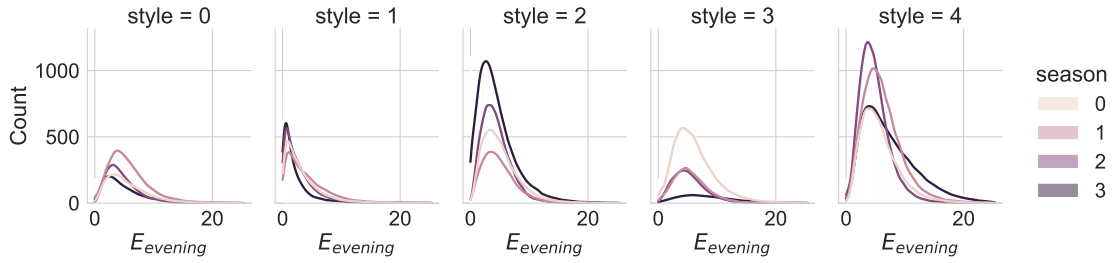


Figure 2.22: Distribution of evening energy use (in KWh) over four seasons for lifestyles

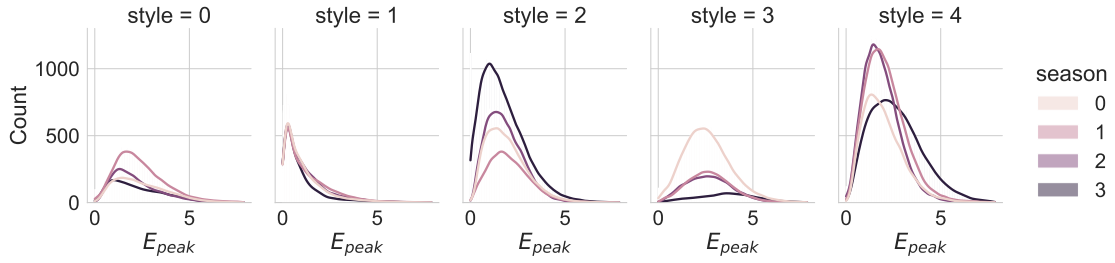


Figure 2.23: Distribution of daily peak energy (in KWh) over four seasons for lifestyles

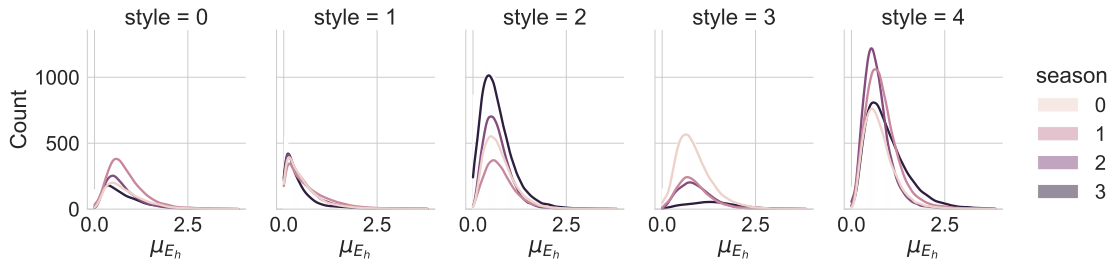


Figure 2.24: Distribution of hourly mean energy (in KWh) over four seasons for lifestyles

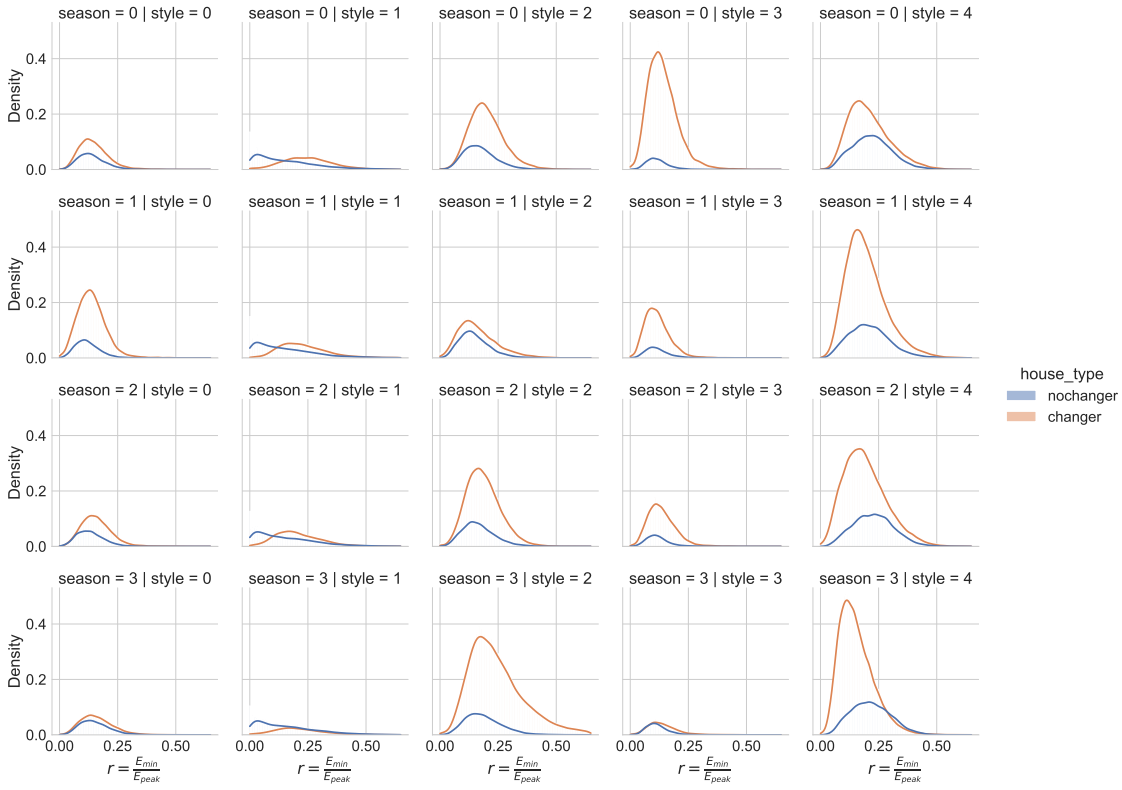


Figure 2.25: Distributions of min (base) to peak energy ratio for Changers and No Changers of five lifestyles over four seasons.

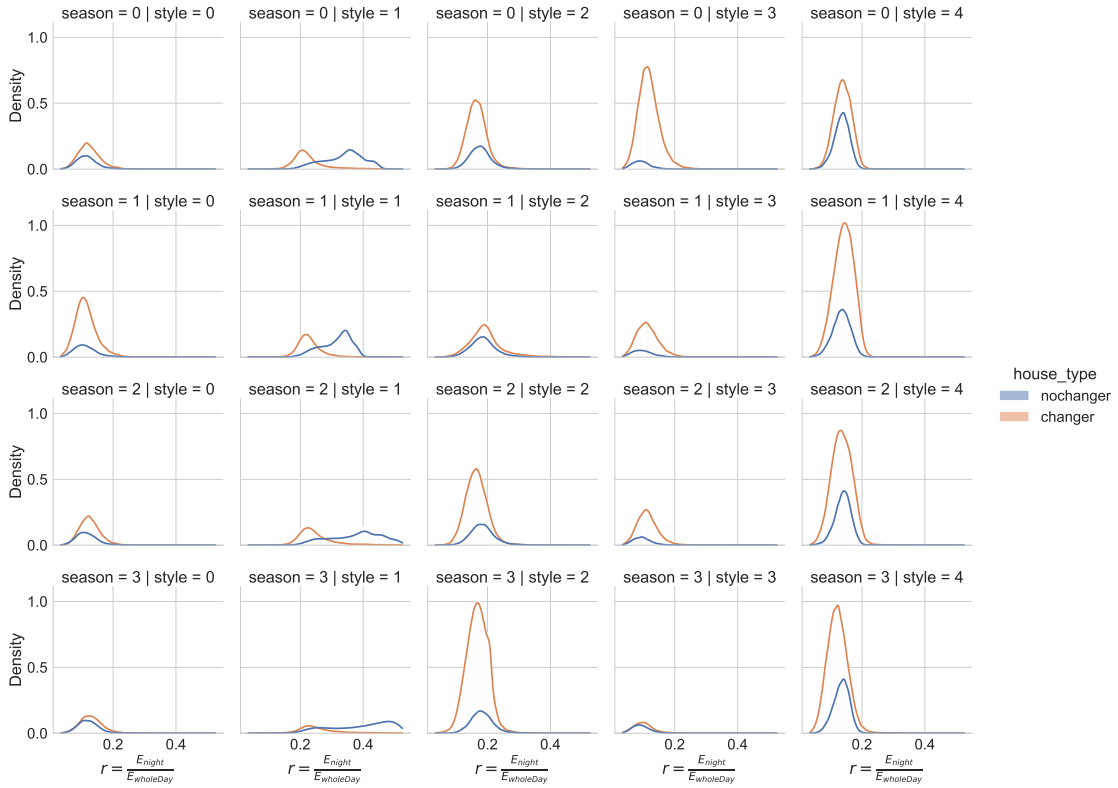


Figure 2.26: Distributions of night to whole day energy ratio for Changers and No Changers of five lifestyles over four seasons.

Table 2.5: Lifestyle classification performance

style index	lifestyle	precision	recall	F1 score
0	active morning	0.7164	0.6315	0.6713
1	night owl	0.8657	0.8176	0.8409
2	everyday is a new day	0.6411	0.6191	0.6299
3	home early	0.6551	0.4685	0.5463
4	home for dinner	0.6652	0.7841	0.7198
5	steady going	0.6417	0.5493	0.5919
				average acc = 0.685

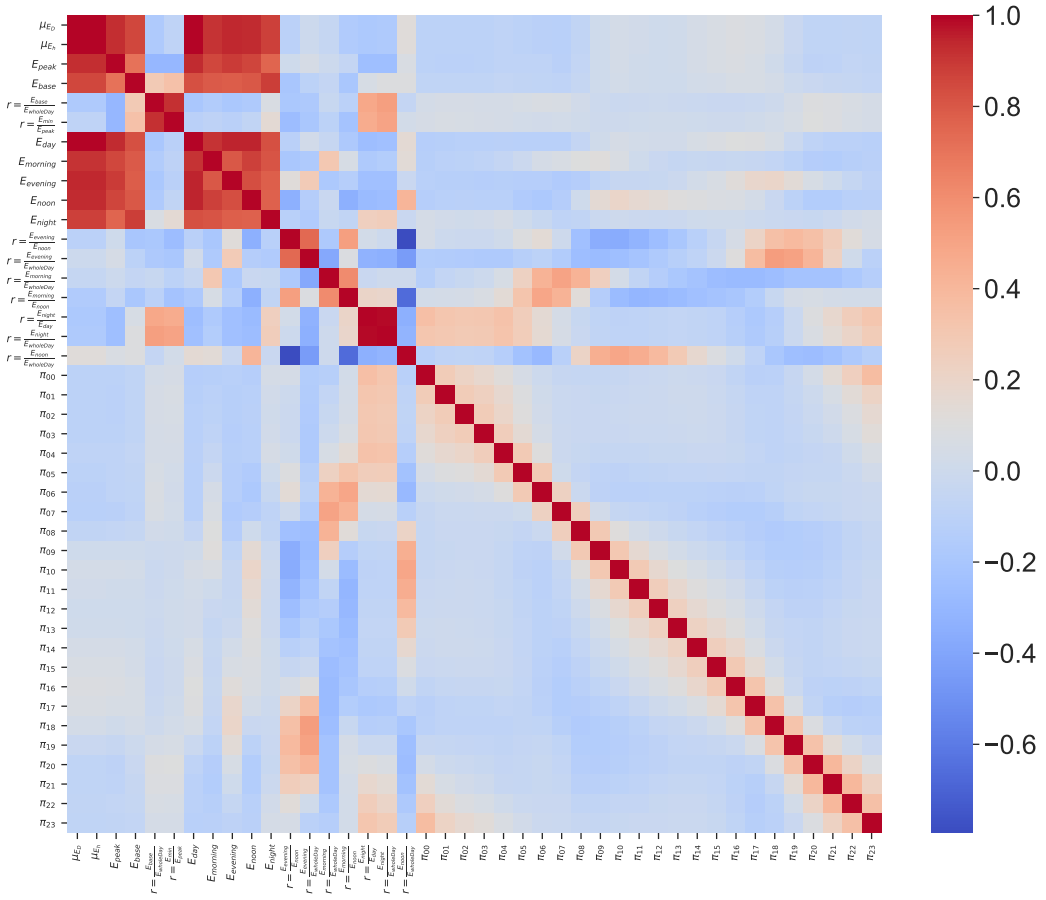
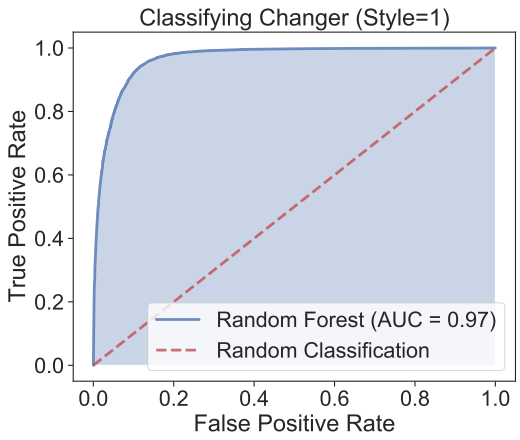


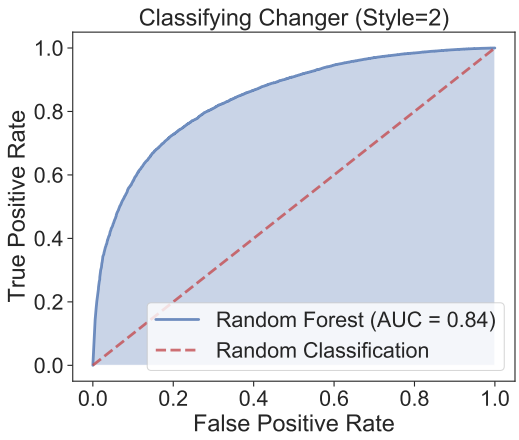
Figure 2.27: Correlation heatmap of features.

than identifying a No Changer because of the higher F1 scores. One exception is the *night owl* lifestyle, which has similar performances of identifying Changers and No Changers given their relatively similar F1 scores. In addition, we show AUC plots identifying Changers vs. No Changers for those five lifestyles (Figure 2.28).

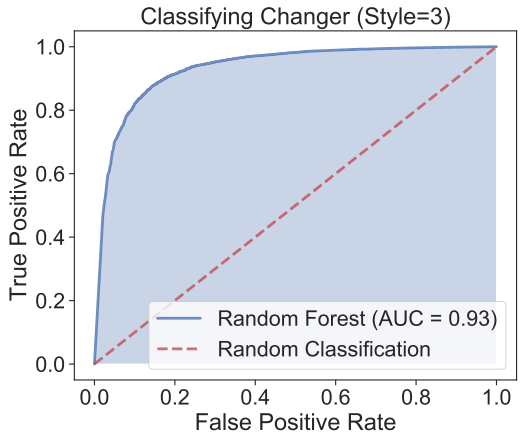
The most important determinants of identifying Changers or No changers are different across the five lifestyles. We present their corresponding top 10 important features in Figure 2.29.



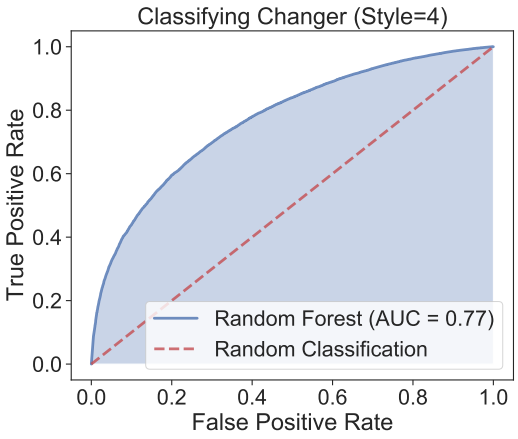
(a) *night owl* style



(b) *everyday is a new day* style



(c) *home early* style



(d) *home for dinner* style

Figure 2.28: AUC of classifying Changer v.s. No Changer.

Table 2.6: active morning lifestyle

label	precision	recall	F1 score
No Changer (0)	0.6481	0.2502	0.3611
Changer (1)	0.8922	0.9786	0.9334
			average acc = 0.879

Table 2.7: night owl lifestyle

label	precision	recall	F1 score
No Changer (0)	0.9301	0.8706	0.8994
Changer (1)	0.9073	0.9508	0.9285
			average acc = 0.917

Table 2.8: everyday is a new day lifestyle

label	precision	recall	F1 score
No Changer (0)	0.6476	0.1533	0.2479
Changer (1)	0.9083	0.9902	0.9475
			average acc = 0.902

Table 2.9: home early lifestyle

label	precision	recall	F1 score
No Changer (0)	0.6966	0.4028	0.5104
Changer (1)	0.9657	0.9897	0.9775
			average acc = 0.957

Table 2.10: home for dinner lifestyle

label	precision	recall	F1 score
No Changer (0)	0.5919	0.1734	0.2682
Changer (1)	0.9657	0.9897	0.9775
			average acc = 0.856

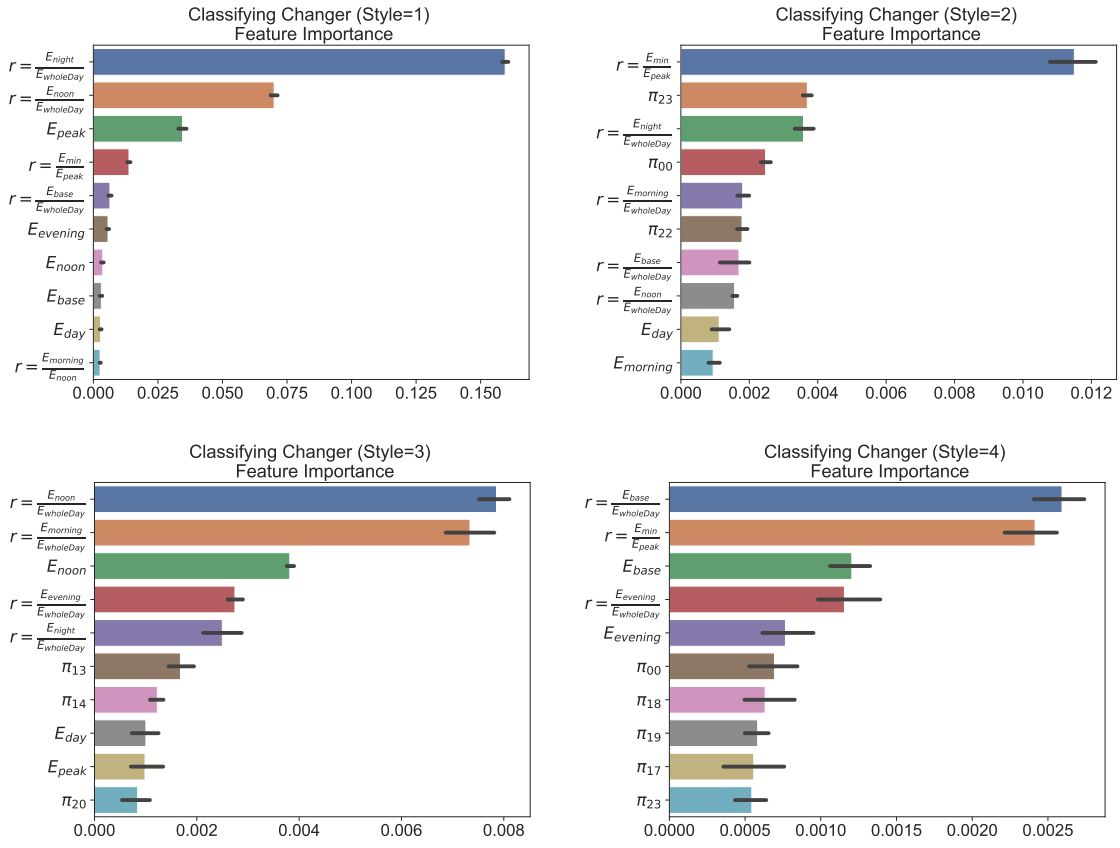


Figure 2.29: Feature importance (Changer v.s. No Changer)

2.5.7 Exploration on influencing features

To verify our identified lifestyles using an alternative approach, we run multinomial logit models to measure how the electricity features influence the (log) odds of having a certain lifestyle among our experimental samples. We treat the steady going lifestyle as the reference group, without loss of the generality, setting it the reference group $Y = 0$. Thus the model is comparing each group outcome with the reference group:

$$\log\left(\frac{P(Y=1)}{P(Y=0)}\right) = \boldsymbol{\theta}_1 \boldsymbol{x}, \cdots, \log\left(\frac{P(Y=5)}{P(Y=0)}\right) = \boldsymbol{\theta}_5 \boldsymbol{x}. \tag{2.13}$$

where $Y = 1, ..., 5$ indicates Active morning, Night owl, Everyday is a new day, Home early, and Home for dinner respectively. The linear coefficients $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_5$ are associated with the lifestyles and the input features are denoted as $\boldsymbol{x}$. Note that we normalize input features between 0 and 1 so that all the scales are in a similar range. We make use of the statsmodels [81] [1] package and use L^1 regularization to fit the parameters and output p-values and generate 95% confidence intervals. The results are averaged values from 150 iterations with a batch size of 5000.

We mainly focus on the features of energy consumption volumes and consumption ratios within a day because they can serve as a loose proxy for features of daily activities. In the following comparison, we consider base energy, base to peak ratio, base energy portion (relative to a day), ratio of evening to whole day, ratio of morning to whole day, ratio of night to whole day, and ratio of noon to whole day, since they are straightforward to harness with daily routines. Table 2.11 to Table 2.15 present these results.

2.5.8 Description of Latent Dirichlet Allocation

In this section, we describe details of Latent Dirichlet Allocation (LDA) and its application in constructing lifestyles. We use the notation listed in Table 2.16.

The LDA model first prescribes K attributes, with each attribute k associated with a distribution ψ_k over shapes in the dictionary. In particular, ψ_k is sampled

[1] https://www.statsmodels.org/stable/index.html

Table 2.11: Multinomial logistic regression results: Active morning.

Param. †	**coef**	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
base_ene	39.9405	16.7984	2.3530	0.122	7.016	72.865
bp_ratio	-12.6192	2.8126	-4.4661	0.001	-18.132	-7.107
base_portion	-0.4703	2.1370	-0.2226	0.449	-4.659	3.718
r_e2w	40.0273	3.7593	10.6363	0.000	32.659	47.395
r_m2w	45.5967	3.8572	11.8167	0.000	38.037	53.157
r_ni2w	-23.5468	4.4598	-5.2854	0.000	-32.288	-14.806
r_no2w	19.0893	3.9245	4.8517	0.001	11.397	26.781
const	-15.8458	2.0431	-7.7429	0.000	-19.850	-11.842

† The parameters from top to bottom rows are base energy, base to peak ratio, base energy portion (relative to a day), ratio of evening to whole day, ratio of morning to whole day, ratio of night to whole day, ratio of noon to whole day, and the constant term.

Table 2.12: Multinomial logistic regression results: Night owl.

Param.	**coef**	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
base_ene	29.3953	13.1345	2.1403	0.126	3.652	55.138
bp_ratio	-2.8727	1.9517	-1.4376	0.255	-6.698	0.953
base_portion	-2.3137	1.9372	-1.1984	0.325	-6.111	1.483
r_e2w	33.0589	4.6545	7.0663	0.000	23.936	42.182
r_m2w	13.5410	4.0855	3.2894	0.031	5.533	21.548
r_ni2w	39.3345	4.7984	8.1428	0.000	29.930	48.739
r_no2w	-12.9092	5.0904	-2.6113	0.130	-22.886	-2.932
const	-13.6910	2.8082	-4.7628	0.001	-19.195	-8.187

Table 2.13: Multinomial logistic regression results: Everyday is a new day.

Param.	**coef**	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
base_ene	-6.2529	12.7474	-0.6382	0.344	-31.237	18.731
bp_ratio	-12.2651	1.8272	-6.6772	0.000	-15.846	-8.684
base_portion	3.2624	1.7237	1.8944	0.165	-0.116	6.641
r_e2w	27.9164	2.5287	11.0360	0.000	22.960	32.873
r_m2w	-10.0360	3.2348	-3.1260	0.029	-16.376	-3.696
r_ni2w	5.4709	2.8215	1.9058	0.131	-0.059	11.001
r_no2w	-3.0341	2.8927	-1.0808	0.346	-8.704	2.636
const	-1.4733	0.5860	-2.4375	0.054	-2.622	-0.325

Table 2.14: Multinomial logistic regression results: Home early.

Param.	**coef**	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
base_ene	63.0495	15.3862	4.0614	0.003	32.893	93.206
bp_ratio	-25.1830	2.7710	-9.0725	0.000	-30.614	-19.752
base_portion	5.0066	2.0665	2.4281	0.115	0.956	9.057
r_e2w	32.1368	3.0962	10.3711	0.000	26.068	38.205
r_m2w	-1.7445	3.4999	-0.5145	0.448	-8.604	5.115
r_ni2w	-24.0826	3.7348	-6.4682	0.000	-31.403	-16.762
r_no2w	30.1940	3.4867	8.6466	0.000	23.360	37.028
const	-6.5285	1.3256	-4.8937	0.000	-9.127	-3.930

Table 2.15: Multinomial logistic regression results: Home for dinner.

Param.	**coef**	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
base_ene	41.2951	12.0066	3.3187	0.016	17.763	64.828
bp_ratio	-5.9587	1.8252	-3.2359	0.028	-9.536	-2.381
base_portion	1.1285	1.7586	0.6460	0.387	-2.318	4.575
r_e2w	50.2073	3.0536	16.4404	0.000	44.222	56.192
r_m2w	2.2251	3.3439	0.6512	0.413	-4.329	8.779
r_ni2w	-13.5150	3.4060	-3.9894	0.006	-20.191	-6.839
r_no2w	19.6216	3.3668	5.8139	0.000	13.023	26.220
const	-11.8697	1.3205	-8.9764	0.000	-14.458	-9.282

Notation	Description
k	Index of attributes (topics)
K	Number of attributes
i	Index of shapes
j	Index of homes or users
α	Dirichlet prior on the attributes in a home
β	Dirichlet prior weight of shapes in a attribute
θ_j	Attribute distribution of home j
θ_{jk}	Proportion of attribute k in home j
ψ_k	Shape distribution of attribute k
ψ_{ki}	Probability of word i occurring in attribute k
s_j	Shape collection of home j
s_{ji}	Shape i in s_j
z_{ji}	Attribute assignment for shape s_{ji} from home j
M	Number of homes
N_j	Number of shapes in home j

Table 2.16: LDA symbol description

from a Dirichlet distribution $Dir(\beta)$. Based on these created attributes, a home j (namely a collection of shapes s_j) is generated by first sampling a distribution θ_j over K attributes from another Dirichlet distribution $Dir(\alpha)$, which determines attribute assignment for each shape in s_j and then choosing each shape s_{ji} based on θ_j. In generating each shape s_{ji}, LDA first samples a particular attribute $z_{ji} \in \{1, ..., K\}$ from multinomial distribution θ_j, and then the shape s_{ji} is selected from a multinomial distribution $\psi_{z_{ji}}$. This process can be summarized in the following steps:

Steps in Latent Dirichlet Allocation

step1: pick shape distribution of each attribute k by $\psi_k \sim Dir(\beta)$

step2: pick attribute distribution for each home j by $\theta_j \sim Dir(\alpha)$

step3: For each home j, for each shape s_{ji} in j:

pick an attribute $z_{ji} \sim \theta_j$;

pick a shape $s_{ji} \sim \psi_{z_{ji}}$

The model fitting can be completed by using variational expectation-maximization

(EM) [82, 54] or Markov Chain Monte Carlo methods (e.g. Gibbs sampling [83]). Both methods can infer the posterior of attribute distribution θ and attribute-shape distribution ψ efficiently. In our experiment, we use sklearn [84] with variational EM algorithm[2] to perform the computation.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

Chapter 3

Generating Private Data with User Customization

Personal devices such as mobile phones can produce and store large amounts of data that can enhance machine learning models; however, this data may contain private information specific to the data owner that prevents the release of the data. We want to reduce the correlation between user-specific private information and the data while retaining the useful information. Rather than training a large model to achieve privatization from end to end, we first decouple the creation of a latent representation, and then privatize the data that allows user-specific privatization to occur in a setting with limited computation and minimal disturbance on the utility of the data. We leverage a Variational Autoencoder (VAE) to create a compact latent representation of the data that remains fixed for all devices and all possible private labels. We then train a small generative filter to perturb the latent representation based on user specified preferences regarding the private and utility information. The small filter is trained via a GAN-type robust optimization that can take place on a distributed device such as a phone or tablet. Under special conditions of our linear filter, we disclose the connections between our generative approach and Rényi differential privacy. We conduct experiments on multiple datasets including MNIST, UCI-Adult, and CelebA, and give a thorough evaluation including visualizing the geometry of the latent embeddings and estimating the empirical mutual information to show the

effectiveness of our approach.

3.1 Introduction

The success of machine learning algorithms relies on not only technical methodologies, but also the availability of large datasets such as images [85]; however, data can often contain sensitive information, such as race or age, that may hinder the owner's ability to release the data to grasp its utility. We are interested in exploring methods of providing privatized data such that sensitive information cannot be easily inferred from the adversarial perspective, while preserving the utility of the dataset. In particular, we consider a setting where many end users are independently gathering data that will be collected by a third party. Each user is incentivized to label their own data with useful information; however, they have the option to create private labels for information that they do not want to share with the database. In the case where data contains a large amount of information such as images, there can be an overwhelming number of potential private and utility label combinations (skin color, age, gender, race, location, medical conditions, etc.). The large number of combinations prevents training a separate method to obscure each set of labels centrally. Furthermore, when participants are collecting data on their personal devices such as mobile phones, they would like to remove private information before the data leaves their devices. Both the large number of personal label combinations coupled with the use of mobile devices requires a privacy scheme to be computationally efficient. In this paper, we propose a method of generating private datasets that makes use of a fixed encoding, thus requiring only a few small neural networks to be trained for each label combination. This approach allows data collecting participants to select any combination of private and utility labels and remove them from the data on their own mobile devices before sending any information to a third party.

In the context of publishing datasets with privacy and utility guarantees, we briefly review a number of similar approaches that have been recently considered, and discuss why they are inadequate at performing in the distributed and customizable setting we have proposed. Traditionally in the domain of generating private datasets, researchers

have made use of differential privacy (DP)[86], which involves injecting certain random noise into the data to prevent the identification of sensitive information [86, 87, 71]. However, finding a globally optimal perturbation using DP may be too stringent of a privacy condition in many high-dimensional data applications. In more recent literature, researchers commonly use Autoencoders [88] to create a compact latent representation of the data, which does not contain private information, but does encode the useful information [89, 90, 91, 92, 93, 14]. A few papers combine strategies involving both DP and Autoencoders [94, 95]; however, all of these recent strategies require training a separate Autoencoder for each possible combination of private and utility labels. Training an Autoencoder for each privacy combination can be computationally prohibitive, especially when working with high dimensional data or when computation must be done on a small local device such as a mobile phone. Therefore, such methods are unable to handle the scenario where each participant must locally train a data generator that obscures their individual choice of private and utility labels. We believe reducing the computation and communication burden is important when dealing with distributed data, because this would be beneficial in many applications such as federated learning [96].

Another line of studies branching on differential privacy focuses on theoretical privacy guarantees of privatization mechanisms. We leverage a previously established relaxation of differential privacy along this line of work, i.e. Rényi differential privacy [97, 98, 99], to determine how much privacy our mechanism can achieve under certain conditions. This notion of differential privacy is weaker than the more common relaxation of (ε, δ)-differential privacy [87].

Primarily, this paper introduces a decoupling of the creation of a latent representation and the privatization of data that allows the privatization to occur in a setting with limited computation and minimal disturbance on the utility of the data. We leverage a generative linear model to create a privatized representation of the data and establish the connection between this simple linear transformation of generative noise with differential privacy. We also build a connection between solving constrained, robust optimization and having Rényi differential privacy under certain conditions. Finally, we run thorough empirical experiments on realistic high-dimensional datasets

with comparison to the related studies. Additionally we contribute a variant on the Autoencoder to improve robustness of the decoder for reconstructing perturbed data and comprehensive investigations into: (i) the latent geometry of the data distribution before and after privatization, (ii) how training against a cross-entropy loss adversary impacts the mutual information between the data and the private label, and (iii) how our linear filter affects the classification accuracy of sensitive labels.

3.2 Problem Statement and Methodology

Inspired by a few recent studies [100, 13, 92], we consider the data privatization as a game between two players: the data generator (data owner) and the adversary (discriminator). The generator tries to inject noise into the data to privatize certain sensitive information contained in the data, while the adversary tries to infer this sensitive information from the data. In order to deal with high dimensional data, we first learn a latent representation or manifold of the data distribution, and then inject noise with specific latent features to reduce the correlation between released data and sensitive information. After the noise has been added to the latent vector, the data can be reconstructed and published without fear of releasing sensitive information. To summarize, the input to our system is the original dataset with both useful and private labels, and the output is a perturbed dataset that has reduced statistical correlation with the private labels, but has maintained information related to the useful labels.

We consider the general setting where a data owner holds a dataset $\mathcal{D}$ that consists of original data X, private/sensitive labels Y, and useful labels U. Thus, each sample i has a record $(x_i, y_i, u_i) \in \mathcal{D}$. We denote the function g as a general mechanism for perturbing the data that enables owner to release the data. The released data is denoted as $\{\tilde{X}, \tilde{U}\}$. Because Y is private information, it won't be released. Thus, for the record i, the released data can be described as $(\tilde{x}_i, \tilde{u}_i) = g(x_i, y_i, u_i)$. We simplify the problem by considering only the case that $\tilde{x}_i = g(x_i, y_i)$[1] for the following description: The corresponding perturbed data $\tilde{X} = g(X, Y)$ and utility attributes U are published

[1]We maintain the U to be unchanged

for use. The adversary builds a learning algorithm h to infer the sensitive information given the released data, i.e. $\hat{Y} = h(\tilde{X})$ where $\hat{Y}$ is the estimate of Y. The goal of the adversary is to minimize the inference loss $\ell(\hat{Y}, Y) = \ell\Big(h\big(g(X,Y)\big), Y\Big)$ on the private labels. Similarly, we denote the estimate of utility labels as $\hat{U} = \nu(\tilde{X}) = \nu(g(X,Y))$. We quantify the utility of the released data through another loss function $\tilde{\ell}$ that captures the utility attributes, i.e. $\tilde{\ell}(\tilde{X}, U) = \tilde{\ell}\Big(\nu\big(g(X,Y)\big), U\Big)$. The data owner wants to maximize the loss that the adversary experiences in order to protect the sensitive information while maintaining the data utility by minimizing the utility loss. Given the previous settings, the data-releasing game can be expressed as follows:

$$\max_{g\in\mathcal{G}} \Big\{ \min_{h\in\mathcal{H}} \mathbb{E}\Big[\ell\Big(h\big(g(X,Y)\big), Y\Big)\Big] - \beta \min_{\nu\in\mathcal{V}} \mathbb{E}\Big[\tilde{\ell}\Big(\nu\big(g(X,Y)\big), U\Big)\Big]\Big\}, \tag{3.1}$$

where β is a hyper-parameter weighing the trade-off between different losses, and the expectation $\mathbb{E}$ is taken over all samples from the dataset. The loss functions in this game are flexible and can be tailored to a specific metric that the data owner is interested in. For example, a typical loss function for classification problems is cross-entropy loss [101]. Because optimizing over the functions g, h, ν is hard to implement, we use a variant of the min-max game that leverages neural networks to approximate the functions. The foundation of our approach is to construct a good posterior approximation for the data distribution in latent space $\mathcal{Z}$, and then to inject context aware noise through a filter in the latent space, and finally to run the adversarial training to achieve convergence, as illustrated in Figure 3.1. Specifically, we consider the data owner playing the generator role that comprises a Variational Autoencoder (VAE) [88] structure with an additional noise injection filter in the latent space. We use θ_g, θ_h, and θ_ν to denote the parameters of neural networks that represent the data owner (generator), adversary (discriminator), and utility learner (util-classifier), respectively. Moreover, the parameters of generator θ_g consists of the encoder parameters θ_e, decoder parameters θ_d, and filter parameters θ_f. The encoder and decoder parameters are trained independently of the privatization process and left fixed, since we decoupled the steps of learning latent representations and generating

the privatized version of the data. Hence, we have

$$\max_{\theta_f} \Big\{ \min_{\theta_h} \mathbb{E}\Big[\ell\Big(h_{\theta_h}\big(g_{\theta_g}(X,Y)\big),Y\Big)\Big] - \beta \min_{\theta_\nu} \mathbb{E}\Big[\tilde{\ell}\Big(\nu_{\theta_\nu}\big(g_{\theta_g}(X,Y)\big),U\Big)\Big]\Big\} \quad (3.2)$$

$$s.t. \qquad D\Big(g_{\theta_e}(X), g_{\theta_f}\big(g_{\theta_e}(X),\epsilon,Y\big)\Big) \leq b, \quad (3.3)$$

where ϵ is standard Gaussian noise, D is a distance or divergence measure, and b is the corresponding distortion budget. The purpose of adding ϵ is to randomize the noise injection generation. The divergence measure captures the closeness of the privatized samples to the original samples while the distortion budget provides a limit on this divergence. The purpose of the distortion budget is to prevent excessive noise injection, which is to avoid deteriorating unspecified information in the data.

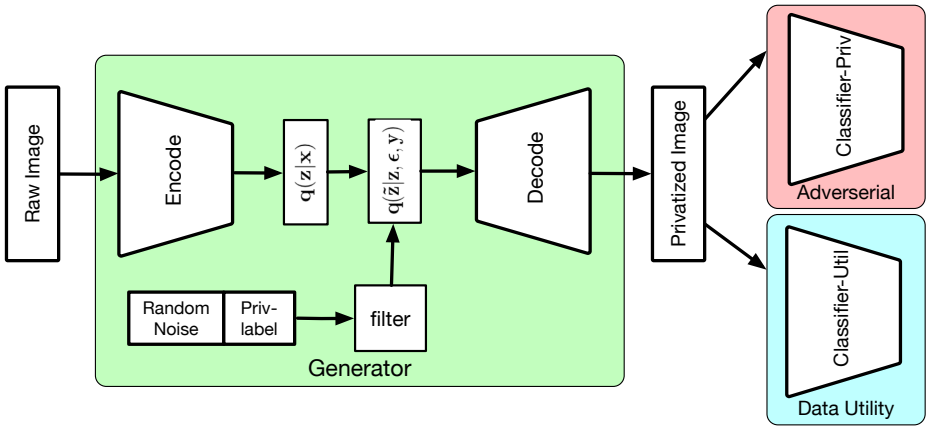


Figure 3.1: Privatization architecture. We decompose the privatization procedure into two steps: 1) training an encoder and decoder; 2) learning a generative filter. The generative filter is learned through a min-max optimization that minimizes utility classifier loss and maximizes adversarial classifier loss.

In principle, we perform the following three steps to complete each experiment. 1)

Train a VAE for the generator without noise injection or min-max adversarial training. Because we want to learn a compact latent representation of the data. Rather than imposing the distortion budget at the beginning, we first train the following objective

$$\begin{aligned} &\min_{\theta_e,\theta_d} -\mathbb{E}_{q(z|x;\theta_e)}[\log p(x|z;\theta_d)] \\ &+ D_{\mathrm{KL}}\big(q(z|x;\theta_e)||p(z)\big), \end{aligned} \tag{3.4}$$

where the posterior distribution $q(z|x;\theta_e)$ is characterized by an encoding network $g_{\theta_e}(x)$, and $p(x|z;\theta_d)$ is similarly the decoding network $g_{\theta_d}(z)$. The distribution $p(z)$ is a prior distribution that is usually chosen to be a multivariate Gaussian for the reparameterization purpose [88]. When dealing with high dimensional data, we develop a variant of the preceding objective that captures three items: the reconstruction loss, KL divergence on latent representations, and improved robustness of the decoder network to perturbations in the latent space (as shown in equation (3.15)). We discuss more details of training a VAE and this variant in section 3.6.2.

2) **Formulate the robust optimization** for min-max GAN-type training [102] with noise injection, which comprises a linear filter,[2] while freezing the weights of the encoder and decoder. In this phase, we instantiate several divergence metrics and various distortion budgets to run our experiments (details in section 3.6.3). When we fix the encoder, the latent variable z can be expressed as $z = g_{\theta_e}(x)$ (or $z \sim q_{\theta_e}(z|x)$ or q_{θ_e} for short), and the new altered latent representation $\tilde{z}$ can be expressed as $\tilde{z} = g_{\theta_f}(z, \epsilon, y)$, where g_{θ_f} represents a filter function (or $\tilde{z} \sim q_{\theta_f}$ for short). The classifiers h and ν can take the latent vector as input when training the filter to reduce the computational burden, as is done in our experiments. We focus on a canonical form for the adversarial training and cast our problem into the following

[2]The filter can be nonlinear such as a small neural network. We focus on the linear case for the remainder of the paper.

robust optimization problem:

$$\min_{\theta_h} \left\{ \max_{q_{\theta_f}} \mathbb{E}_{q_{\theta_f}} \big(\ell(h_{\theta_h}; \tilde{z})\big) - \beta \mathbb{E}_{q_{\theta_f}} \big(\tilde{\ell}(\nu_{\theta_\nu}; \tilde{z})\big) \right\} \tag{3.5}$$

$$\text{s.t.} \quad D_f(q_{\theta_f} || q_{\theta_e}) \leq b \tag{3.6}$$

$$\theta_\nu = \arg\min_{\theta_\nu} \mathbb{E}_{q_{\theta_f}} \big(\tilde{\ell}(\nu_{\theta_\nu}; \tilde{z})\big), \tag{3.7}$$

where D_f is the *f-divergence* in this case. We omit the default condition of $\mathbb{E}q_{\theta_f} = 1$ for the simplicity of the expressions. By decomposing and analyzing some special structures of our linear filter, we disclose the connection between our generative approach and Rényi differential privacy in Theorem 3 and Proposition 2 (in appendix section 3.6.4).

3) **Learn the adaptive classifiers** for both the adversary and utility labels according to the data $[\tilde{X}, Y, U]$ $\big($or $[\tilde{Z}, Y, U]$ if the classifier takes the latent vector as input$\big)$, where the perturbed data $\tilde{X}$ (or $\tilde{Z}$) is generated based on the trained generator. We validate the performance of our approach by comparing metrics such as classification accuracy and empirical mutual information. Furthermore, we visualize the geometry of the latent representations, e.g. figure 3.4, to give intuitions behind how our framework achieves privacy.

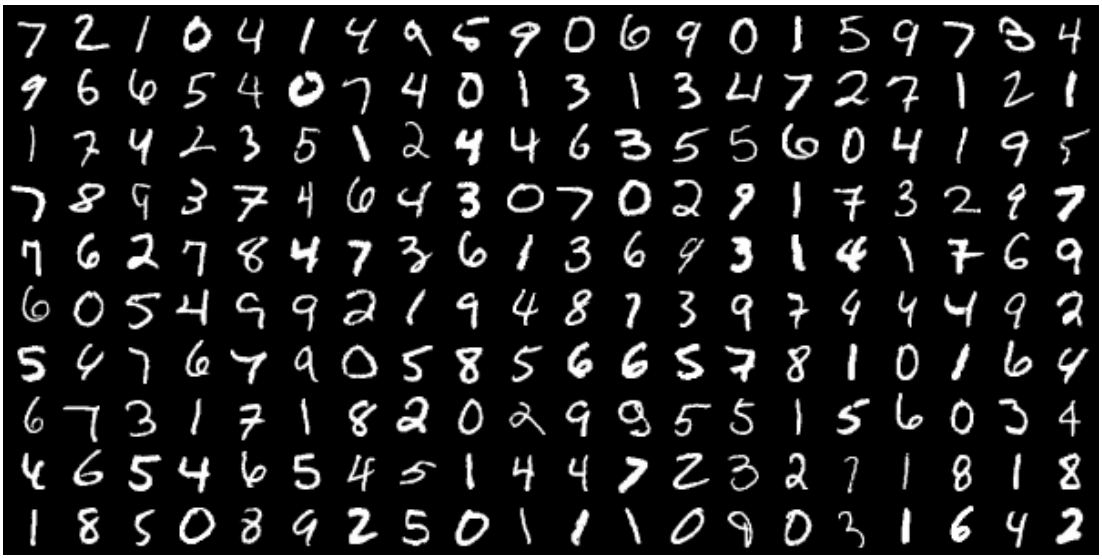
3.3 Experiments and Results

We verified our idea on four datasets. The first is the MNIST dataset [103] of handwritten digits, commonly used as a toy example in machine learning literature. We have two cases involving this dataset: In MNIST Case 1, we preserve information regarding whether the digit contains a circle (i.e. digits 0,6,8,9), but privatize the value of the digit itself. In MNIST Case 2, we preserve information on the parity (even or odd digit), but privatize whether or not the digit is greater than or equal to 5. Figure 3.2 shows a sample of the original dataset along with the same sample perturbed to remove information on the digit identity but maintain the digit as a circle-containing digit. The input to the algorithm is the original dataset with labels, while the output is the perturbed data as shown. The second experimental dataset is

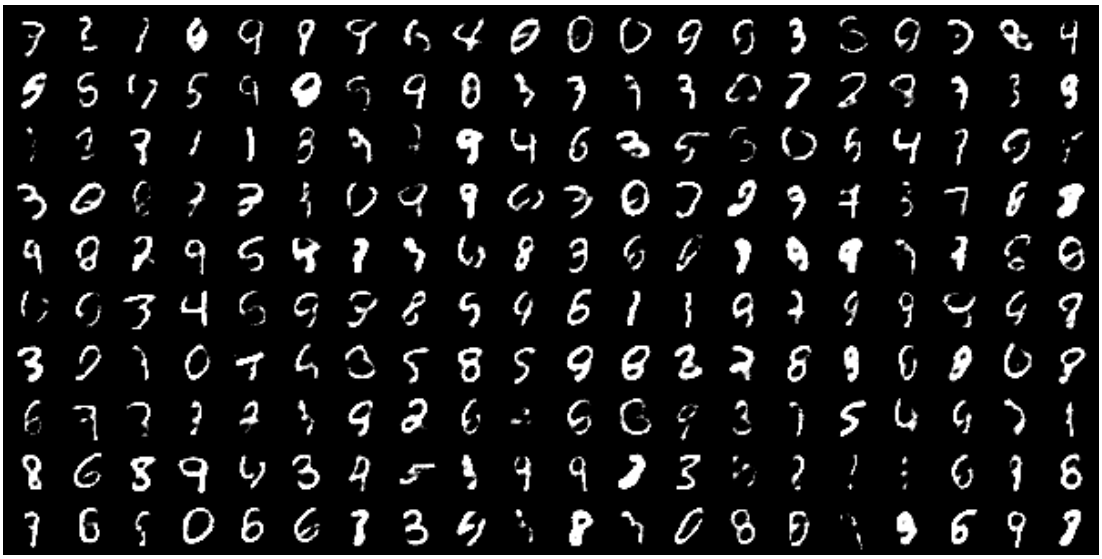
the UCI-adult income dataset [104] that has 45222 anonymous adults from the 1994 US Census. We preserve whether an individual has an annual income over \$50,000 or not while privatizing the gender of that individual. The third dataset is UCI-abalone [105] that consists of 4177 samples with 9 attributes including target labels. The fourth dataset is the CelebA dataset [106] containing color images of celebrity faces. For this realistic example, we preserve whether the celebrity is smiling, while privatizing many different labels (gender, age, etc.) independently to demonstrate our capability to privatize a wide variety of labels with a single latent representation.

MNIST Case 1: We considered the digit value itself as the private attribute and the digit containing a circle or not as the utility attribute. Figure 3.2 shows samples of this case. Specific classification results before and after privatization are given in the form of confusion matrices in Figures 3.3a and 3.3b, demonstrating a significant reduction in private label classification accuracy. These results are supported by our illustrations of the latent space geometry in Figure 3.4 obtained via uniform manifold approximation and projection (UMAP) [107]. Specifically, figure 3.4b shows a clear separation between circle digits (on the right) and non-circle digits (on the left). We also investigated the sensitivity of classification accuracy for both labels with respect to the distortion budget (for KL-divergence) in Figure 3.3c, demonstrating that increasing the distortion budget rapidly decreases the private label accuracy while maintaining the accuracy of utility label. We also compare these results to a baseline method based on an additive Gaussian mechanism (discussed in section 3.6.4), and we found that this additive Gaussian mechanism performs worse than our generative adversarial filter in terms of keeping the utility and protecting the private labels because the Gaussian mechanism yields lower utility and worse privacy (i.e. an adversary can have higher prediction accuracy of private labels) than the min-max generative filter approach. In appendix section 3.6.4, we show our min-max generative filter approach can connect to Rényi differential privacy under certain conditions.

MNIST Case 2: This case has the same setting as the experiment given in [108] where we consider odd or even digits as the target utility and large (≥ 5) or small (< 5) value as the private label. Rather than training a generator based on a fixed classifier, as done in [108], we take a different modeling and evaluation approach that

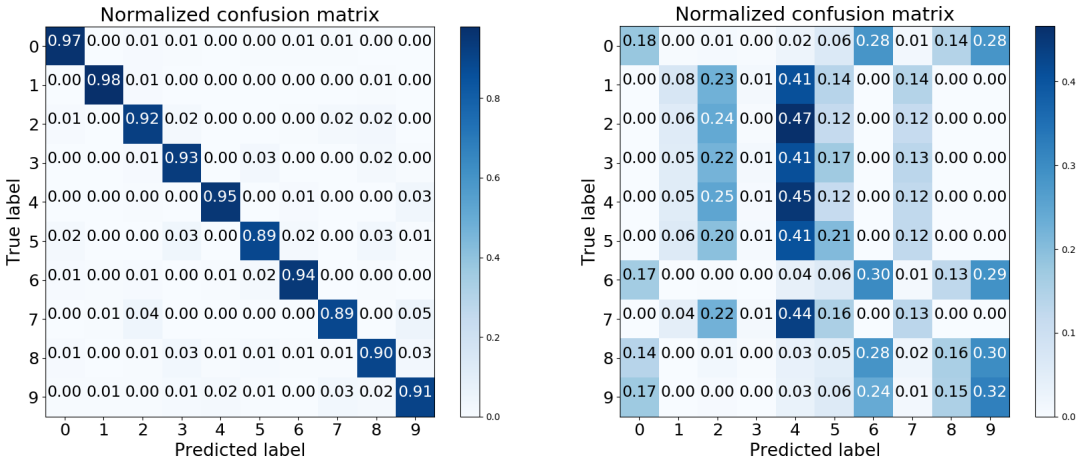


(a) Sample of original images



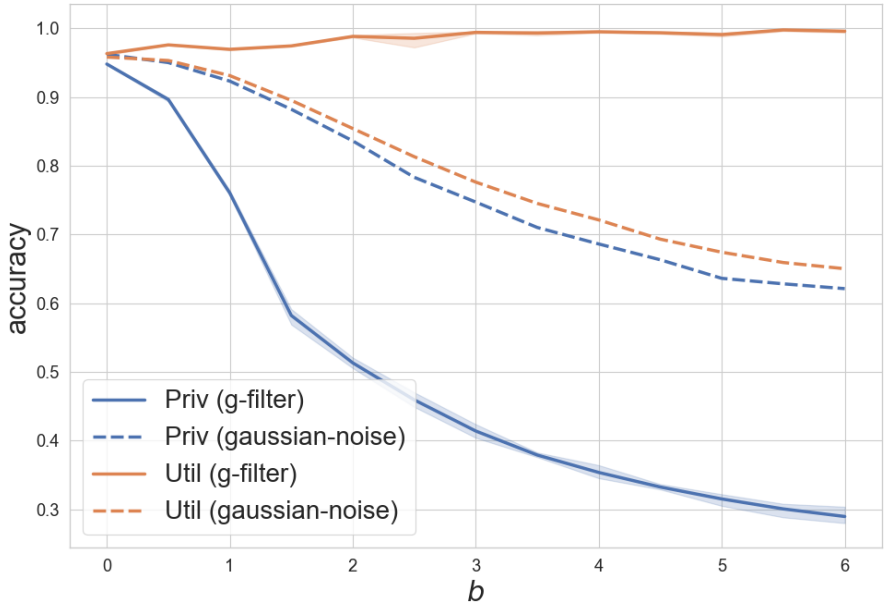
(b) Same images perturbed to privatize digit ID

Figure 3.2: Visualization of digits pre- and post-noise injection and adversarial training. We find that digit IDs are randomly switched while circle digits remain circle digits and non-circle digits remain as non-circle digits.



(a) raw

(b) privatized



(c) classification accuracy

Figure 3.3: Classifying digits in MNIST. Original digits can be easily classified with more than 90% accuracy on average, yet the new perturbed digits have a significantly lower accuracy as expected. Specifically, many circle digits are incorrectly classified as other circle digits and similarly for the non-circle digits. Figure 3.3c demonstrates that classification accuracy on the private label decreases quickly while classification on the utility label remains nearly constant as the distortion budget increases. Our approach is superior to the baseline Gaussian mechanism based on the senario of adding equivalent noise on each coordinates.

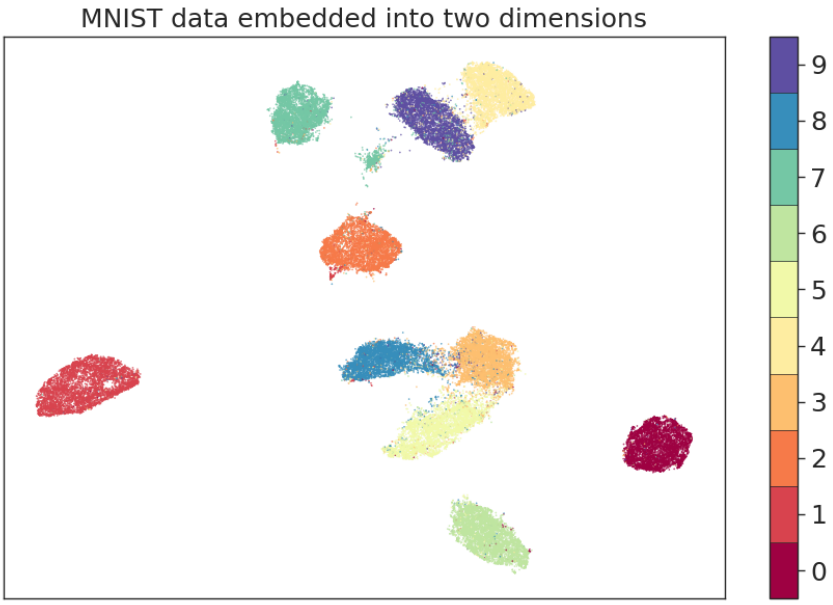
allows the adversarial classifier to update dynamically. We find that the classification accuracy of the private attribute drops down from 95% to 65% as the distortion budget grows. Meanwhile our generative filter doesn't deteriorate the target utility too much, maintaining a classification accuracy above 87% for the utility label as the distortion increases, as shown in figure 3.7b. We discuss more results in the appendix section 3.6.8, together with results verifying the reduction of mutual information between the data and the private labels.

To understand whether specifying the target utility is too restrictive for other usage of the data, we conducted the following experiment. Specifically, we measure the classification accuracy of the circle attribute from case 1, while using the privatization scheme from case 2 (preserving digit parity). This tests how the distortion budget prevents excessive loss of information of non-target attributes. When training for case 2, the circle attribute from case 1 is not included in the loss function by design; however, as seen in Table 3.1, the classification accuracy on the circle is not more diminished than the target attribute (odd). A more detailed plot of the classification accuracy can be found in Figure 3.7c in the appendix section 3.6.8. This experiment demonstrates that the privatized data maintains utility beyond the predefined utility labels used in training.

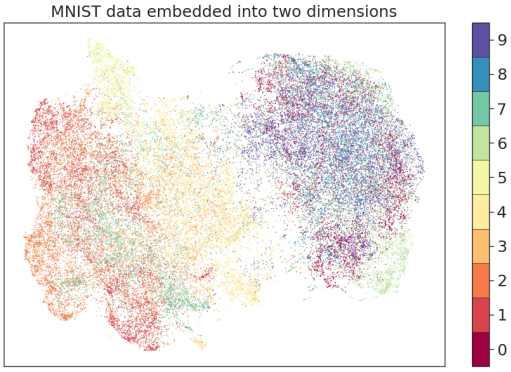
Table 3.1: Accuracy of private label (≥ 5), target label (odd), and non-target label (circle) for MNIST dataset. The raw embedding yielded by the VAE before privacy is denoted as *emb-raw*. The embedding yielded from the generative filter after privacy is denoted as *emb-g-filter*.

Data	**Priv. attr.**	**Util. attr.**	**Non-tar. attr.**
emb-raw	0.951	0.952	0.951
emb-g-filter	0.687	0.899	0.9

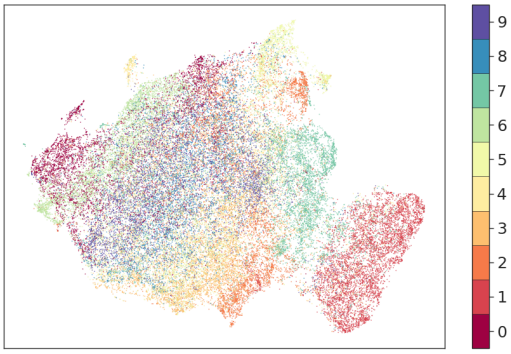
UCI-Adult: We conduct the experiment on this dataset by setting the private label to be gender and the utility label to be income. All the data is preprocessed to binary values for the ease of training. We compare our method with the models of Variational Fair AutoEncoder (VFAE)[100] and Lagrangian Mutual Information-based Fair Representations (LMIFR) [93]. The corresponding accuracy and area-under receiver operating characteristic curve (AUROC) of classifying private label



(a) raw



(b) moderate perturbation



(c) heavy perturbation

Figure 3.4: Visualization of the latent geometry. The original embedding in Figure 3.4a is clearly segmented into individual clusters for each digit; however, when we allow a distortion budget of $b = 1.5$, as shown in Figure 3.4b, the digits are separated according to the circle or non-circle property by the gap between the left and right clouds of points. A larger distortion budget ($b = 5$) nests all samples close together with some maintained local clusters as seen in Figure 3.4c.

and utility label are shown in Table 3.2. Our method has the lowest accuracy and the smallest AUROC on the privatized gender attribute. Although our method doesn't perform best on classifying the utility label, it still achieves comparable results in terms of both accuracy and AUROC, which are described in Table 3.2.

Table 3.2: Accuracy (acc.) and Area-Under-ROC (auroc.) of private label (gender) and target label (income) for UCI-Adult dataset.

Model	**Private attr.**		**Utility attr.**	
	acc.	auroc.	acc.	auroc.
VAE [88]	0.850 ± 0.007	0.843 ± 0.007	0.837 ± 0.009	0.755 ± 0.006
VFAE [100]	0.802 ± 0.009	0.703 ± 0.013	**0.851 ± 0.007**	0.761 ± 0.011
LMIFR [93]	0.728 ± 0.014	0.659 ± 0.012	0.829 ± 0.009	0.741 ± 0.013
Ours (w. generative filter)	**0.717 ± 0.008**	0.632 ± 0.011	0.822 ± 0.011	0.731 ± 0.015

UCI-Abalone: We partition the rings label into two classes based on if individual had less or more than 10 rings. Such a setup follows the same setting in [109]. We treat the rings label as the utility label. For the private label, we pick sex because we hope classifying rings could be less correlated with the sex of abalones. There are three categories in sex: male, female and infant. We see that having small amount of distortion budget (b=0.01) reduces the classification accuracy of private label significantly. However, the accuracy and auroc remain around the similar level comparing with the raw data, unless we have large distortion budget (b=10), shown in Table 3.3.

Table 3.3: Accuracy (acc.) of both the private label (sex) and utility label (rings), and the Area-Under-ROC (auroc.) of utility label for the Abalone dataset.

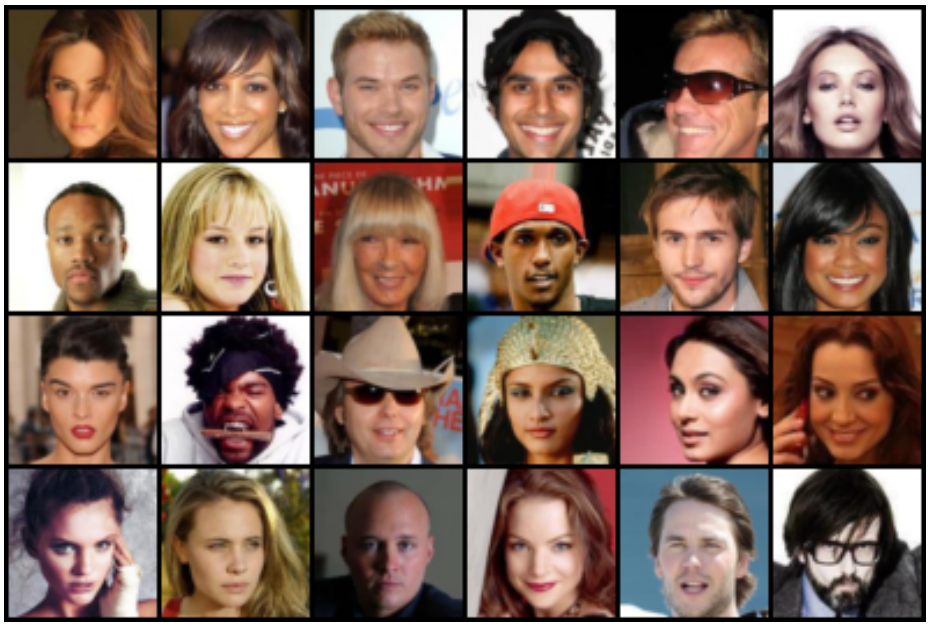
b	**priv. attr.** acc.	**util. attr.** acc.	**util. attr.** auroc.
0 (raw)	0.546 ± 0.011	0.748 ± 0.016	0.75 ± 0.003
0.01	0.321 ± 0.007	0.733 ±0.010	0.729 ±0.003
0.1	0.314 ± 0.010	0.721 ± 0.012	0.728 ±0.003
1	0.313 ± 0.02	0.720 ± 0.010	0.729 ±0.006
10	0.305 ± 0.033	0.707 ± 0.011	0.699 ±0.009

CelebA: For the CelebA dataset, we consider the case when there exist many

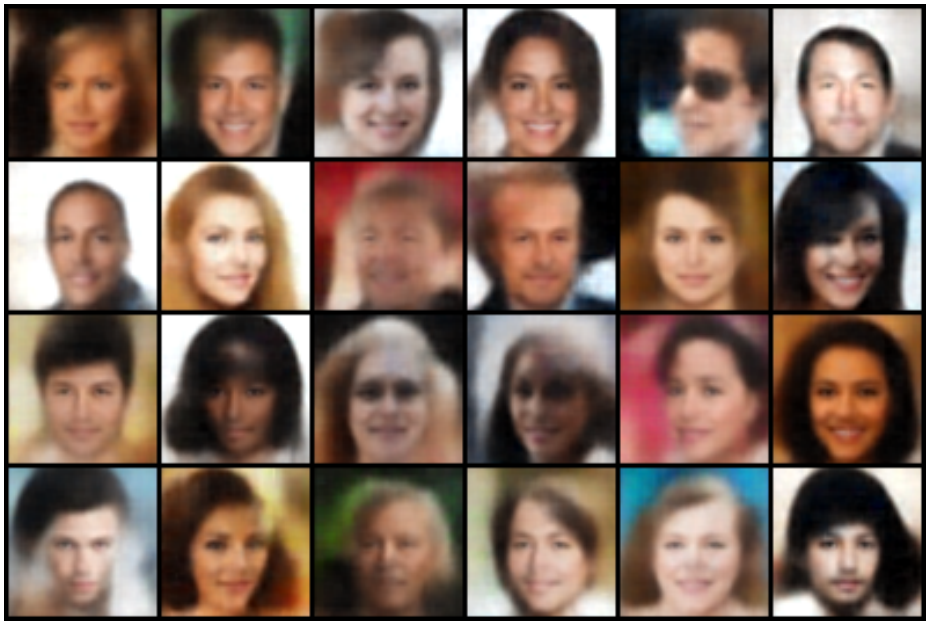
private and utility label combinations depending on the user's preferences. Specifically, we experiment on the private labels gender (male), age (young), attractive, eyeglasses, big nose, big lips, high cheekbones, or wavy hair, and we set the utility label as smiling for each private label to simplify the experiment. Table 3.4 shows the classification results of multiple approaches. Our trained generative adversarial filter reduces the accuracy down to 73% on average, which is only 6% more than the worst case accuracy demonstrating the ability to protect the private attributes. Meanwhile, we only sacrifice a small amount of the utility accuracy (3% drop), which ensures that the privatized data can still serve the desired classification tasks. (All details are summarized in Table 3.4.) We show samples of the gender-privatized images in Figure 3.5, which indicates the desired phenomenon that some female images are switched into male images and some male images are changed into female images. More example images on other privatized attributes, including eyeglasses, wavy hair, and attractive, can be found in appendix section 3.6.9.

Table 3.4: Classification accuracy on CelebA. The row **VAE-emb** is our initial classification accuracy on the latent vectors from our trained encoder. The row **Random-guess** demonstrates the worst possible classification accuracy. The row **VAE-g-filter** is the classification accuracy on the perturbed latent vectors yielded by our generative adversarial filters. The state-of-the-art classifier [110] can achieve 87% accuracy on the listed private labels on average while our trained VAE can achieve a comparable accuracy (83% on average) on the more compact latent representations. More importantly, our trained generative adversarial filter can reduce the accuracy down to 73% on average, which is only 6% more than the worst case accuracy demonstrating the ability to protect the private attributes. Meanwhile, we only sacrifice a small amount of the utility accuracy (3% drop), which ensures that the privatized data can still serve the desired classification tasks

	Private attr.									**Utility attr.**
	Male	Young	Attractive	H. Cheekbones	B. lips	B. nose	Eyeglasses	W. Hair	**Avg**	Smiling
[106]	0.98	0.87	0.81	0.87	0.68	0.78	0.99	0.80	0.84	0.92
[110]	0.98	0.89	0.82	0.87	0.73	0.83	0.99	0.84	0.87	0.93
VAE-emb	0.90	0.84	0.80	0.85	0.68	0.78	0.98	0.78	0.83	0.86
Random-guess	0.51	0.76	0.54	0.54	0.67	0.77	0.93	0.63	0.67	0.50
VAE-g-filter	0.61	0.76	0.62	0.78	0.67	0.78	0.93	0.66	0.73	0.83



(a) raw samples



(b) same samples gender privatized

Figure 3.5: Sampled images. We privatize gender while keeping the expression (smiling or not) as the utility label. The generative filter switches some female faces to male faces and also some male faces to female faces while preserving the celebrity's smile. The blurriness of the privatized images is because of the compactness of the latent representation that is generated from the VAE model, which is a normal occurrence among VAE models, and not from our privatization scheme. More details can be found in figure 3.10 in section 3.6.9.

3.4 Discussion

Distributed training with customized preferences: In order to clarify how our scheme can be run in a local and distributed fashion, we performed a basic experiment on the MNIST dataset with 2 independent users to demonstrate this capability. The first user adopts the label of digit ≥ 5 as private and odd or even as the utility label. The second user prefers the opposite and wishes to privatize odd or even and maintain ≥ 5 as the utility label. Details on the process can be found in appendix section 3.6.7.

The VAE is trained separately by a data aggregator and the parameters are handed to each user. Then, each user learns a separate linear generative filter that privatizes their data. Since the linear generative filter is trained on the low dimensional latent representation, it is small enough for computation on a personal device. After privatization, we can evaluate the classification accuracy on the private and utility labels as measured by adversaries trained on the full aggregated privatized dataset, which is the combination of each users privatized data. Table 3.5 demonstrates the classification accuracy on the two users privatized data. This shows how multiple generative filters can be trained independently using a single encoding to successfully privatize small subsets of data.

Table 3.5: Accuracy of adversarial classifiers on two users private labels

Classifier type	**User 1 (privatize ≥ 5)**	**User 2 (privatize odd)**
Private attr.	0.679	0.651
Utility attr.	0.896	0.855

Connection to Rényi differential privacy: We bridge the connection between our linear privatization generator and traditional differential privacy mechanisms through the following high level descriptions (proofs and details are thoroughly explained in appendix section 3.6.4). To begin, we introduce the Rényi-divergence and a relaxation of differential privacy based on this divergence called Rényi differential privacy [99] [see Definition 3]. We then in Theorem 3 provide the specifications under which our linear filter provides (ε, α)-Rényi differential privacy. Finally, in Proposition 2 we harness our robust optimization to the Rényi differential privacy.

In appendix section 3.6.6, we also establish reasons why our privatization filter is able to decrease the classification accuracy of the private attributes.

3.5 Conclusion and Future Work

In this paper, we proposed an architecture for privatizing data while maintaining the utility that decouples for use in a distributed manner. Rather than training a very deep neural network or imposing a particular discriminator to judge real or fake images, we first trained VAE that can create a comprehensive low dimensional representation from the raw data. We then found smart perturbations in the low dimensional space according to customized requirements (e.g. various choices of the private label and utility label), using a robust optimization approach. Such an architecture and procedure enables small devices such as mobile phones or home assistants (e.g. Google home mini) to run a light weight learning algorithm to privatize data under various settings on the fly. We demonstrated that our proposed additive noise method can be Rényi differentially private under certain conditions and compared our results to a traditional additive Gaussian mechanism for differential privacy.

Finally, we discover an interesting connection between our idea of decorrelating the released data with sensitive attributes and the notion of learning fair representations[111, 100, 92]. In fairness learning, the notion of demographic parity requires the prediction of a target output to be independent with respect to certain protected attributes. We find our generative filter could be used to produce a fair representation after privatizing the raw embedding, which shares a similar idea to that of demographic parity. Proving this notion along with other notions in fairness learning such as equal odds and equal opportunity [112] will be left for future work.

3.6 Supplements

3.6.1 Related work

We introduce several papers [94, 100, 13, 113, 114] which are closely related to our ideas with several distinctions. [94] presents a minimax filter without a decoder or a

distortion constraint. It's not a generative model. The presented simulation focuses on low dimensional and time series data instead of images, so our specific model architecture, loss functions, and training details are quite different. [100] proposed a Variational Fair Autoencoder that requires training from end to end, which is computationally expensive with high dimensional data, many privacy options, and training on edge devices. They also use Maximum Mean Discrepancy (MMD) to restrict the distance between two samples. We decouple the approach into a VAE and a linear filter, while adopting the f-divergence (equivalent to α-divergence in our context) to constrain the distance between latent representations. One benefit of doing that is such a divergence has connections to Rényi differential privacy under certain conditions. [13] focuses on one-dimensional variables and uses a synthetic dataset for the simulation, which remains unclear how it can be scaled up to a realistic dataset. Many studies do not recover the privatized data to the original dimension from the latent representation to give qualitative visual support, whereas our experiments conduct a thorough evaluation through checking the latent representation and reconstructing images back to the original dimension. [113] uses a variant of the classical GAN objective. They require the generator to take a target alternative label to privatize the original sensitive label, which leads to deterministic generation. Instead of lumping all losses together and training deep models from end to end, we decouple the encoding/decoding and noise injection without requiring a target alternative label. [114] proposes a framework that creates fair representation via disentangling certain labels. Although the work builds on the VAE with modifications to factorize the attributes, it focuses on the sub-group fair classification (e.g. similar false positive rate among sub-groups) rather than creating privacy-preserving data. Furthermore, we have two discriminators: private and utility. In addition to the VAE, we use KL-divergence to control the sample distortion and minimax robust optimization to learn a simple linear model. Through this, we disclose the connection to the Rényi differential privacy, which is also a new attempt.

3.6.2 VAE training

A variational autoencoder is a generative model defining a joint probability distribution between a latent variable z and original input x. We assume the data is generated from a parametric distribution $p(x|z;\theta)$ that depends on latent variable z, where θ are the parameters of a neural network that is usually a decoder net. Maximizing the marginal likelihood $p(x|z;\theta)$ directly is usually intractable. Thus, we use the variational inference method proposed by [88] to optimize $\log p(x|z;\theta)$ over an alternative distribution $q(z|x;\phi)$ with an additional KL divergence term $D_{\mathrm{KL}}(q(z|x;\phi)||p(z))$, where the ϕ are parameters of a neural net and $p(z)$ is an assumed prior over the latent space. The resulting cost function is often called evidence lower bound (ELBO)

$$\log p(x;\theta) \geq \mathbb{E}_{q(z|x;\phi)}[\log p(x|z;\theta)] - D_{\mathrm{KL}}\big(q(z|x;\phi)||p(z)\big) = \mathcal{L}_{\mathrm{ELBO}}. \tag{3.8}$$

Maximizing the ELBO is implicitly maximizing the log-likelihood of $p(\mathbf{x})$. The negative objective (also known as negative ELBO) can be interpreted as minimizing the reconstruction loss of a probabilistic autoencoder and regularizing the posterior distribution towards a prior. Although the loss of the VAE is mentioned in many studies [88, 100], we include the derivation of the following relationships for context:

$$\begin{aligned} &D_{\mathrm{KL}}(q(z)||p(z|x;\theta)) \\ &= \sum_z q(z)\log\frac{q(z)}{p(z|x;\theta)} = -\sum_z q(z)\log p(z,x;\theta) + \log p(x;\theta) \underbrace{-H(q)}_{\sum_z q(z)\log q(z)} \geq 0 \end{aligned} \tag{3.9}$$

.

The evidence lower bound (ELBO) for any distribution q has the following property:

$$\log p(x;\theta) \geq \sum_z q(z)\log p(z,x;\theta) - \sum_z q(z)\log q(z) \quad [\text{since } D_{\mathrm{KL}}(\cdot,\cdot)\geq 0] \tag{3.10}$$

$$= \sum_z q(z)\Big(\log p(z,x;\theta) - \log p(z)\Big) - \sum_z q(z)\Big(\log q(z) - \log p(z)\Big) \tag{3.11}$$

$$\overset{\text{(i)}}{=} \mathbb{E}_{q(z|x;\phi)}[\log p(x|z;\theta)] - D_{\mathrm{KL}}\big(q(z|x;\phi)||p(z)\big) = \mathcal{L}_{\mathrm{ELBO}}, \tag{3.12}$$

where equality (i) holds because we treat encoder net $q(z|x;\phi)$ as the distribution $q(z)$. By placing the corresponding parameters of the encoder and decoder networks and the negative sign on the ELBO expression, we get the loss function equation (3.4). The architecture of the encoder and decoder for the MNIST experiments is explained in section 3.6.7.

In our experiments with the MNIST dataset, the negative ELBO objective works well because each pixel value (0 black or 1 white) is generated from a Bernoulli distribution. However, in the experiments of CelebA, we change the reconstruction loss into the ℓ_p norm of the difference between the raw and reconstructed samples because the RGB pixels are not Bernoulli random variables, but real-valued random variables between 0 and 1. We still add the regularization KL term as follows:

$$\mathbb{E}_{x' \sim p(x|z;\theta)}\big[||x - x'||_{\mathrm{p}}\big] + \gamma D_{\mathrm{KL}}\big(q(z|x;\phi)||p(z)\big). \tag{3.13}$$

Throughout the experiments we use a Gaussian $\mathcal{N}(0, I)$ as the prior $p(z)$, x is sampled from the data, and γ is a hyper-parameter. The reconstruction loss uses the ℓ_2 norm by default because it is widely adopted in image reconstruction, although the ℓ_1 norm is acceptable too.

When training the VAE, we additionally ensure that small perturbations in the latent space will not yield huge deviations in the reconstructed space. More specifically, we denote the encoder and decoder to be g_e and g_d respectively. The generator g can be considered as a composition of an encoding and decoding process, i.e. $g(X) = (g_d \circ g_e)(X) = g_d(g_e(X))$, where we ignore the Y inputs here for the purpose of simplifying the explanation. One desired intuitive property for the decoder is to maintain that small changes in the input latent space $\mathcal{Z}$ still produce plausible faces similar to the original latent space when reconstructed. Thus, we would like to impose some Lipschitz continuity property on the decoder, i.e. for two points $z^{(1)}, z^{(2)} \in \mathcal{Z}$, we assume $||g_d(z^{(1)}) - g_d(z^{(2)})|| \leq C_L||z^{(1)} - z^{(2)}||$ where C_L is some Lipschitz constant (or equivalently $||\nabla_z g_d(z)|| \leq C_L$). In the implementation of our experiments, the

gradient for each batch (with size m) is

$$\nabla_z g_d(z) = \begin{bmatrix} \frac{\partial g_d(z^{(1)})}{\partial z^{(1)}} \\ \vdots \\ \frac{\partial g_d(z^{(m)})}{\partial z^{(m)}} \end{bmatrix}. \tag{3.14}$$

It is worth noticing that $\frac{\partial g_d(z^{(i)})}{\partial z^{(i)}} = \frac{\partial \sum_{i=1}^{m} g_d(z^{(i)})}{\partial z^{(i)}}$, because $\frac{\partial g_d(z^{(j)})}{\partial z^{(i)}} = 0$ when $i \neq j$. To avoid the iterative calculation of each gradient within a batch, we define Z as the batched latent input, $Z = \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(m)} \end{bmatrix}$ and use $\nabla_z g_d(z) = \frac{\partial}{\partial Z} \sum_{i=1}^{m} g_d(z^{(i)})$. The loss used for training the VAE is modified to be

$$\mathbb{E}_{x' \sim p(x|z;\theta_d)}\big[||x - x'||_2\big] + \gamma D_{\mathrm{KL}}\big(q(z|x;\theta_e)||p(z)\big) + \kappa \mathbb{E}_{z \sim r_\alpha(z)}\Big[\big(||\nabla_z(g_{\theta_d}(z)||_2 - C_L\big)_+^2\Big], \tag{3.15}$$

where x are samples drawn from the image data, γ, κ, and C_L are hyper-parameters, and $(x)_+$ means $\max\{x, 0\}$. Finally, r_α is defined by sampling $\alpha \sim \mathsf{Unif}[0, 1]$, $Z_1 \sim p(z)$, and $Z_2 \sim q(z|x)$, and returning $\alpha Z_1 + (1-\alpha)Z_2$. We optimize over r_α to ensure that points in between the prior and learned latent distribution maintain a similar reconstruction to points within the learned distribution. This gives us the Lipschitz continuity properties we desire for points perturbed outside of the original learned latent distribution.

3.6.3 Robust optimization and adversarial training

In this section, we formulate the generator training as a robust optimization. Essentially, the generator is trying to construct a new latent distribution that reduces the correlation between data samples and sensitive labels while maintaining the correlation with utility labels by leveraging the appropriate generative filters. The new latent distribution, however, cannot deviate from the original distribution too much (bounded by b in equation (3.3)) to maintain the general quality of the reconstructed images. To simplify the notation, we will use h for the classifier h_{θ_h} (a similar notion

applies to ν or ν_{θ_ν}). We also consider the triplet $(\tilde{z}, y, u)$ as the input data, where $\tilde{z}$ is the perturbed version of the original embedding z, which is the latent representation of image x. The values y and u are the sensitive label and utility label respectively. Without loss of generality, we succinctly express the loss $\ell(h; (\tilde{z}, y))$ as $\ell(h; \tilde{z})$ [similarly expressing $\tilde{\ell}(\nu; (\tilde{z}, u))$ as $\tilde{\ell}(\nu; \tilde{z})$]. We assume the sample input $\tilde{z}$ follows the distribution q_ψ that needs to be determined. Thus, the robust optimization is

$$\min_h \max_{q_\psi} \mathbb{E}_{q_\psi}\left[\ell(h; \tilde{z})\right] + \beta \min_{q_\psi} \min_\nu \mathbb{E}_{q_\psi}\left[\tilde{\ell}(\nu; \tilde{z})\right] \tag{3.16}$$

$$\text{s.t.} \qquad D_f(q_\psi || q_\phi) \leq b, \tag{3.17}$$

where q_ϕ is the distribution of the raw embedding z (also known as $q_{\theta_e}(z|x)$). In the constraint, the f-divergence [115, 116] between q_ψ and q_ϕ is $D_f(q_\psi||q_\phi) = \int q_\phi(z) f(\frac{q_\psi(z)}{q_\phi(z)}) d\mu(z)$ (assuming q_ψ and q_ϕ are absolutely continuous with respect to measure μ). A few typical divergences [117], depending on the choices of f, are

1. KL-divergence $D_{\text{KL}}(q_\psi||q_\phi) \leq b$, by taking $f(t) = t \log t$
2. reverse KL-divergence $D_{\text{KL}}(q_\phi||q_\psi) \leq b$, by taking $f(t) = -\log t$
3. χ^2-divergence $D_{\chi^2}(q_\psi||q_\phi) \leq b$, by taking $f(t) = \frac{1}{2}(t-1)^2$.

In the remainder of this section, we focus on the KL and χ^2 divergence to build a connection between the divergence based constraint we use and norm-ball based constraints seen in [118, 119, 120], and [121].

When we run the constrained optimization, for instance in terms of KL-divergence, we make use of the Lagrangian relaxation technique [122] to put the distortion budget as the penalty term in the loss of objective as

$$\min \left\{ \lambda_1 \max\{D_{\text{KL}} - b, 0\} + \lambda_2 (D_{\text{KL}} - b)^2 \right\}. \tag{3.18}$$

Although this term is not necessarily convex in model parameters, the point-wise max and the quadratic term are both convex in D_{KL}. Such a technique is often used in many constrained optimization problems in the context of deep learning or GAN related work [123].

Extension to multivariate Gaussian

When we train the VAE in the beginning, we impose the distribution q_ϕ to be a multivariate Gaussian by penalizing the KL-divergence between $q_{\theta_e}(z|x)$ and a prior normal distribution $\mathcal{N}(0, I)$, where I is the identity matrix. Without loss of generality, we can assume the raw encoded variable z follows a distribution q_ϕ that is the Gaussian distribution $\mathcal{N}(\mu_1, \Sigma_1)$ (more precisely $\mathcal{N}\big(\mu_1(x), \Sigma_1(x)\big)$, where the mean and variance depends on samples x, but we suppress the x to simplify notation). The new perturbed distribution q_ψ is then also a Gaussian $\mathcal{N}(\mu_2, \Sigma_2)$. Thus, the constraint for the KL divergence becomes

$$
\begin{aligned}
&D_{\mathrm{KL}}(q_\psi||q_\phi) = \mathbb{E}_{q_\psi}(\log \frac{q_\psi}{q_\phi}) \\
&= \frac{1}{2}\mathbb{E}_{q_\psi}[-\log\det(\Sigma_2) - (z-\mu_2)^T\Sigma_2^{-1}(z-\mu_2) + \log\det(\Sigma_1) + (z-\mu_1)^T\Sigma_1^{-1}(z-\mu_1)] \\
&= \frac{1}{2}\big[\log\frac{\det(\Sigma_1)}{\det(\Sigma_2)} - \underbrace{d}_{=\mathrm{Tr}(\mathbf{I})} + \mathrm{Tr}(\Sigma_1^{-1}\Sigma_2) + (\mu_1-\mu_2)^T\Sigma_1^{-1}(\mu_1-\mu_2)\big] \le b.
\end{aligned}
$$

To further simplify the scenario, we consider the case that $\Sigma_2 = \Sigma_1 = \Sigma$, then

$$
D_{\mathrm{KL}}(q_\psi||q_\phi) = \frac{1}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) = \frac{1}{2}||\mu_1-\mu_2||^2_{\Sigma^{-1}} \le b. \tag{3.19}
$$

When $\Sigma = I$, the preceding constraint is equivalent to $\frac{1}{2}||\mu_1-\mu_2||_2^2$. It is worth mentioning that such a divergence-based constraint is also connected to the norm-ball based constraint on samples.

In the case of χ^2-divergence,

$$
\begin{aligned}
&D_{\chi^2}(q_\psi||q_\phi) = E_{q_\phi}[\frac{1}{2}(\frac{q_\psi}{q_\phi}-1)^2] \\
&= \frac{1}{2}\Bigg[\frac{\det(\Sigma_1)}{\det(\Sigma_2)}\exp\big(-\mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + \underbrace{d}_{=\mathrm{Tr}(\mathbf{I})} - \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 + 2\mu_1^T\Sigma_2^{-1}\mu_2\big) \\
&- 2\frac{\det(\Sigma_1)^{\frac{1}{2}}}{\det(\Sigma_2)^{\frac{1}{2}}}\exp\big(-\frac{1}{2}\mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + \frac{d}{2} - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 - \frac{1}{2}\mu_2^T\Sigma_2^{-1}\mu_2 + \mu_1^T\Sigma_2^{-1}\mu_2\big) + 1\Bigg].
\end{aligned}
$$

When $\Sigma_1 = \Sigma_2 = \Sigma$, we have the following simplified expression

$$\begin{aligned} D_{\chi^2}(q_\psi||q_\phi) &= \frac{1}{2}\Big[\exp\big(-||\mu_1-\mu_2||^2_{\Sigma^{-1}}\big) - 2\exp\big(-\frac{1}{2}||\mu_1-\mu_2||^2_{\Sigma^{-1}}\big) + 1\Big] \\ &= \frac{1}{2}[e^{-2s} - 2e^{-s} + 1] = \frac{1}{2}(e^{-s}-1)^2 \end{aligned}$$

where $s = \frac{1}{2}||\mu_1-\mu_2||^2_{\Sigma^{-1}}$. Letting $\frac{1}{2}(e^{-s}-1)^2 \leq b$ indicates $-\sqrt{2b} \leq (e^{-s}-1) \leq \sqrt{2b}$. Since the value of s is always non negative as a norm, $s \geq 0 \implies e^{-s} - 1 \leq 0$. Thus, we have $s \leq -\log\big((1-\sqrt{2b})_+\big)$. Therefore, when the divergence constraint $D_{\chi^2}(q_\psi||q_\phi) \leq b$ is satisfied, we have $||\mu_1-\mu_2||^2_{\Sigma^{-1}} \leq -2\log\big((1-\sqrt{2b})_+\big)$, which is similar to equation (3.19) with a new constant for the distortion budget.

We use these relationships in our implementation as follows. We define μ to be functions that split the last layer of the output of the encoder part of the pretrained VAE, $g_{\theta_e}(x)$, and take the first half as the mean of the latent distribution. We let σ be a function that takes the second half portion to be the diagonal values of the variance of the latent distribution. Then, our implementation of $||\mu\big(g_{\theta_e}(x)\big) - \mu\big(g_{\theta_f}(g_{\theta_e}(x), w, y)\big)||^2_{\sigma(g_{\theta_e}(x))^{-1}} \leq b$ is equivalent to $||\mu_1-\mu_2||^2_{\Sigma^{-1}} \leq b$. As shown in the previous two derivations, optimizing over this constraint is similar to optimizing over the defined KL and χ^2-divergence constraints.

3.6.4 Comparison and connection to differential privacy

The basic intuition behind differential privacy is that it is very hard to tell whether the released sample $\tilde{z}$ originated from raw sample x or x' (or z vs. z'), thus, protecting the privacy of the raw samples. Designing such a releasing scheme, which is also often called a channel or mechanism, usually requires some randomized response or noise perturbation. Although the goal does not necessarily involve reducing the correlation between released data and sensitive labels, it is worth investigating the notion of differential privacy and comparing the performance of a typical scheme to our setting because of its prevalence in privacy literature. Furthermore, we establish how our approach can be Rényi differentially private with certain specifications. In this section, we use the words channel, scheme, mechanism, and filter interchangeably as they have the same meaning in our context. Also, we overload the notation of ε

and δ because they are the conventions in differential privacy literature.

Definition 1 $\big[(\varepsilon, \delta)$*-differential privacy [71]* $\big]$ *Let* $\varepsilon, \delta \geq 0$. *A channel* Q *from space* $\mathcal{X}$ *to output space* $\mathcal{Z}$ *is differentially private if for all measurable sets* $S \subset \mathcal{Z}$ *and all neighboring samples* $\{x_1, \ldots, x_n\} = x_{1:n} \in \mathcal{X}$ *and* $\{x'_1, \ldots, x'_n\} = x'_{1:n} \in \mathcal{X}$,

$$Q(Z \in S | x_{1:n}) \leq e^{\varepsilon} Q(Z \in S | x'_{1:n}) + \delta. \tag{3.20}$$

An alternative way to interpret this definition is that with high probability $1 - \delta$, we have the bounded likelihood ratio $e^{-\varepsilon} \leq \frac{Q(z|x_{1:n})}{Q(z|x'_{1:n})} \leq e^{\varepsilon}$ (The likelihood ratio is close to 1 as ε goes to 0)[3]. Consequently, it is difficult to tell if the observation z is from x or x' if the ratio is close to 1. In the following discussion, we consider the classical Gaussian mechanism $\tilde{z} = f(z) + w$, where f is some function (or query) that is defined on the latent space and $w \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. We first include a theorem from [71] to disclose how differential privacy using the Gaussian mechanism can be satisfied by our baseline implementation and constraints in the robust optimization formulation. We denote a pair of neighboring inputs as $z \simeq z'$ for abbreviation.

Theorem 1 $\big[$*[71] Theorem A.1*$\big]$ *For any* $\varepsilon, \delta \in (0, 1)$, *the Gaussian mechanism with parameter* $\sigma \geq \frac{L\sqrt{2\log(\frac{1.25}{\delta})}}{\varepsilon}$ *is* (ε, δ)*-differential private, where* $L = \max_{z \simeq z'} ||f(z) - f(z')||_2$ *denotes the* l_2*-sensitivity of* f.

Next, we introduce a relaxation of differential privacy that is based on the Rényi divergence.

Definition 2 $\big[$*Rényi divergence ([99], Definition 3)*$\big]$. *Let* P *and* Q *be distributions on a space* $\mathcal{X}$ *with densities* p *and* q *(with respect to a measure* μ*). For* $\alpha \in [1, \infty]$, *the Rényi-*α*-divergence between* P *and* Q *is*

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \int \left(\frac{p(x)}{q(x)}\right)^{\alpha} q(x) d\mu(x), \tag{3.21}$$

where the values $\alpha \in \{1, \infty\}$ *are defined in terms of their respective limits.*

[3] Alternatively, we may express it as the probability $P(|\log \frac{Q(z|x_{1:n})}{Q(z|x'_{1:n})}| > \varepsilon) \leq \delta$

In particular, $\lim_{\alpha\downarrow 1} D_\alpha(P||Q) = D_{\mathrm{KL}}(P||Q)$. We use Rényi divergences because they satisfy $\exp\big((\alpha-1)D_\alpha(P||Q)\big) = D_f(P||Q)$ when f-divergence is defined by $f(t) = t^\alpha$. And such an equivalent relationship has a natural connection with the f-divergence constraint in our robust optimization formulation. With this definition, we introduce the Rényi differential privacy, which is a strictly stronger relaxation than the (ε, δ)-differential privacy relaxation.

Definition 3 [*Rényi differential privacy ([99], Definition 4)*]. *Let $\varepsilon \geq 0$ and $\alpha \in [1, \infty]$. A mechanism F from $\mathcal{X}^n$ to output space $\mathcal{Z}$ is (ε, α)- Rényi private if for all neighboring samples $x_{1:n}, x'_{1:n} \in \mathcal{X}^n$,*

$$D_\alpha\big(F(\cdot|x_{1:n})||F(\cdot|x'_{1:n})\big) \leq \varepsilon. \tag{3.22}$$

For the basic Gaussian mechanism, we apply the additive Gaussian noise on z directly to yield

$$\tilde{z} = z + w, \quad w \sim \mathcal{N}(0, \sigma^2 I). \tag{3.23}$$

We first revisit the basic Gaussian mechanism and its connection to Rényi-differential privacy.

Theorem 2 *Let $L = \max_{z,z' \in \mathcal{Z}} ||z - z'||_2, \forall z, z'$ and $w \sim \mathcal{N}(0, \sigma^2 I)$. Then the basic Gaussian mechanism shown in equation (3.23) is $(\varepsilon, \alpha)-$Rényi private if $\sigma^2 = \frac{\alpha L^2}{2\varepsilon}$.*

Proof of Theorem 2. Considering two examples z and z', we calculate the Rényi divergence between $(z + w) \sim \mathcal{N}(z, \sigma^2 I)$ and $(z' + w) \sim \mathcal{N}(z', \sigma^2 I)$:

$$D_\alpha(\mathcal{N}(z, \sigma^2 I)||\mathcal{N}(z', \sigma^2 I)) \overset{\text{(i)}}{=} \frac{\alpha}{2\sigma^2}||z - z'||_2^2 \leq \frac{\alpha}{2\sigma^2} \max_{z,z' \in \mathcal{Z}} ||z - z'||_2^2 = \frac{\alpha L^2}{2\sigma^2} = \varepsilon. \tag{3.24}$$

The equality (i) is shown in the deferred proofs in equation (3.57). Arranging $\sigma^2 = \frac{\alpha L^2}{2\varepsilon}$ gives the desired result. Although this result is already known as Lemma 2.5 in [98], we simplify the proof technique.

Because normal distributions are often used to approximate data distributions in real applications, we now consider the scenario where the original data z is distributed as a multivariate normal $\mathcal{N}(\mu, \Sigma)$. This is a variant of additive Gaussian mechanism, as we add noise to the mean, i.e.

$$\tilde{\mu} = \mu + \sigma w, \quad w \sim \mathcal{N}(0, I). \tag{3.25}$$

We have the following proposition.

Proposition 1 *Suppose a dataset Z with sample z is normally distributed with parameters $\mathcal{N}(\mu, \Sigma)$, and there exists a scalar σ^2 such that $\max_{z,z' \in Z} ||z - z'|| \leq \sigma^4 \operatorname{Tr}(\Sigma^{-1})$. Under the simple additive Gaussian mechanism, when both $\mathbb{E}\big[D_\alpha\big(\mathcal{N}(\mu + \sigma w, \Sigma) || \mathcal{N}(\mu, \Sigma)\big)\big] \leq b$, then such a mechanism satisfies (b, α) Rényi differential privacy.*

Proof of Proposition 1:

$$b \overset{\text{(use assump.)}}{\geq} \mathbb{E}\Big[D_\alpha\big(\mathcal{N}(\mu + \sigma w, \Sigma) || \mathcal{N}(\mu, \Sigma)\big)\Big] \tag{3.26}$$

$$= \mathbb{E}\Big[\frac{\alpha}{2}(\mu + \sigma w - \mu)^T \Sigma^{-1} (\mu + \sigma w - \mu)\Big] = \frac{\alpha}{2}\sigma^2 \mathbb{E}[w^T \Sigma^{-1} w] \tag{3.27}$$

$$= \frac{\alpha\sigma^2}{2} \mathbb{E}[\operatorname{Tr}(\Sigma^{-1} w w^T)] = \frac{\alpha\sigma^2}{2} \operatorname{Tr}(\mathbb{E}[\Sigma^{-1} w w^T]) \tag{3.28}$$

$$= \frac{\alpha\sigma^2}{2} \operatorname{Tr}\Big(\mathbb{E}[\Sigma^{-1}] \underbrace{\mathbb{E}[w w^T]}_{I}\Big) = \frac{\alpha\sigma^2}{2} \operatorname{Tr}(\Sigma^{-1}) \tag{3.29}$$

$$\overset{\text{(use assump.)}}{\geq} \frac{\alpha\sigma^2}{2} \frac{1}{\sigma^4} \max_{z,z' \in Z} ||z - z'|| = \frac{\alpha}{2\sigma^2} \max_{z,z' \in Z} ||z - z'|| \geq D_\alpha(\mathcal{N}(z, \Sigma) || \mathcal{N}(z', \Sigma)) \tag{3.30}$$

With a prescribed budget b, the predetermined divergence α, and a known data covariance Σ, we can learn an adjustable σ. Moreover, if $\sigma^4 \geq \frac{\max ||z - z'||}{\operatorname{Tr}(\Sigma^{-1})}$ and $b \geq \frac{\alpha\sigma^2}{2} \operatorname{Tr}(\Sigma^{-1})$, we have (b, α) Rényi differential privacy.

To build the connection between the Rényi differential privacy and our constrained robust optimization, we explicitly impose some matrix properties of our linear filter. We denote the matrix Γ to be the linear filter, the one-hot vector y_s to represent private label y, and w to be standard Gaussian noise. This method can be considered

as a linear filter mechanism, a variant of additive transformed Gaussian mechanism. The output $\tilde{z}$ generated from z and w can be described as

$$\tilde{z} = z + \Gamma \begin{bmatrix} w \\ y_s \end{bmatrix} = z + \begin{bmatrix} A & V \end{bmatrix} \begin{bmatrix} w \\ y_s \end{bmatrix} = z + Aw + Vy_s. \tag{3.31}$$

We decompose the matrix Γ into two parts: A controls the generative noise and V controls the bias. Now we present the following theorem.

Theorem 3 *Let matrix* Γ *be decomposed into* $\Gamma = \begin{bmatrix} A & V \end{bmatrix}$, $\sigma_{min} > 0$ *be the minimum eigenvalue of* AA^T, $||V||_1 = \tau$, *and* $L = \max_{z,z' \in \mathcal{Z}} ||z - z'||_2$, *then the mechanism*

$$\tilde{z} = z_{1:n} + Aw + Vy_{s\{1:n\}} \tag{3.32}$$

satisfies (ε, α)*-Rényi privacy, where* $\varepsilon = \frac{\alpha d}{2\sigma_{min}}(L^2 + 2\tau^2)$ *and* $z_{1:n}, y_{s\{1:n\}}$ *are* d*-dimensional samples.*

Proof of Theorem 3. Consider two examples (z, y_s) and (z', y'_s). Because $w \sim \mathcal{N}(0, I)$, the corresponding output distributions yielded from these two examples are $\mathcal{N}(z + Vy_s, AA^T)$ and $\mathcal{N}(z' + Vy'_s, AA^T)$ through the linear transformation of Gaussian random vector w. The resulting Rényi divergence is

$$\begin{aligned}
& D_\alpha\big(\mathcal{N}(z + Vy_s, AA^T) || \mathcal{N}(z' + Vy'_s, AA^T)\big) && \text{(3.33a)} \\
& \overset{\text{(i)}}{=} \frac{\alpha}{2}(z - z' + Vy_s - Vy'_s)^T (AA^T)^{-1}(z - z' + Vy_s - Vy'_s) && \text{(3.33b)} \\
& \overset{\text{(ii)}}{\leq} \frac{\alpha d}{2\sigma_{min}} ||z - z' + Vy_s - Vy'_s||_2^2 && \text{(3.33c)} \\
& \overset{\text{(iii)}}{\leq} \frac{\alpha d}{2\sigma_{min}} \big\| z - z' + 2|v_{j^*}| \big\|_2^2 && \text{(3.33d)} \\
& \overset{\text{(iv)}}{\leq} \frac{\alpha d}{2\sigma_{min}} (||z - z'||_2^2 + 2||V||_1^2) && \text{(3.33e)} \\
& \leq \frac{\alpha d}{2\sigma_{min}} (\max_{z,z \in \mathcal{Z}} ||z - z'||_2^2 + 2||V||_1^2) && \text{(3.33f)}
\end{aligned}$$

where equality (i) uses the property of Rényi divergence between two Gaussian distributions [124] that is provided in the deferred proof of equation (3.57), and inequality (ii) uses the assumption that σ_{min} is the minimum eigenvalue of AA^T (so that $AA^T \geq \sigma_{min}I$). Because y_s is a one-hot vector, Vy_s returns a particular column of V where the column index is aligned with the index of non-zero entry of y_s. Inequality (iii) picks the index j^* from column vectors $[v_1, ...v_j, ..., v_K] = V$ (if the label y has K classes) such that $j^* = \arg\max_j \sum_i |v_{ij}|$. We denote $|v_{j^*}|$ as a vector taking absolute value of each entry in v_{j^*}. The final (iv) simply applies the triangle inequality and $\max_j \sum_i |v_{ij}| = ||V||_1$ (maximum absolute column sum of the matrix). The remainder of the proof is straightforward by changing $||z - z'||_2^2$ to $\max_{z,z\in\mathcal{Z}} ||z - z'||_2^2$ in equation (3.33e) as the mechanism runs through all samples. Finally, setting $\frac{\alpha d}{2\sigma_{min}}(\max ||z-z'||_2^2 + 2||V||_1^2) = \varepsilon$ with substitutions of L and τ gives the desired result.

Consequently, we connect the α-Rényi divergence to differential privacy by presenting Corollary 4.

Corollary 4 *The* $\left(\frac{\alpha d}{2\sigma_{min}}(L^2+2\tau^2), \alpha\right)$ *Rényi differential privacy mechanism also satisfies* $\left(\frac{\alpha d}{2\sigma_{min}}(L^2 + 2\tau^2) + \frac{1}{\alpha-1}\log\frac{1}{\delta}, \delta\right)$ *differential privacy for any* $\delta > 0$.

Proof of Corollary 4. The proof is an immediate result of proposition 3 in Minronov's work [99] when we set $\varepsilon = \frac{\alpha d}{2\sigma_{min}}(L^2 + 2\tau^2)$.

Remark: When having high privacy with small ε, we set σ_{min} to be large. This can be seen from the following relationship.

$$D_f = \exp\left((\alpha-1)D_\alpha\right) \overset{\text{(theorem 3)}}{\leq} \exp\left((\alpha-1)\frac{\alpha d(L^2+2\tau^2)}{2\sigma_{min}}\right) \overset{\text{(def)}}{\leq} b \tag{3.34a}$$

$$\implies \sigma_{min} \geq \left[\frac{\alpha(\alpha-1)d(L^2+2\tau^2)}{2\log b}\right]_+. \tag{3.34b}$$

One intuitive relationship is that if L is large (i.e. $\max_{z,z'\in\mathcal{Z}} ||z - z'||_2$ is large), the σ_{min} goes up with a quadratic growth rate with respect to L. Because large L indicates z is significantly distinct from z', which requires a large variance of the noise

(i.e. large AA^T) to obfuscate the original samples, σ_{min} also shrinks with logarithmic growth of b when the privacy level is less stringent.

However we also notice that the eigenvalues of AA^T cannot grow to infinity, as shown in Proposition 2.

Proposition 2 *Suppose a dataset $\mathcal{Z}$ with sample z, a d-dimensional vector, is normally distributed with parameters $\mathcal{N}(\mu, \Sigma)$, and there exists A and V such that $\mathrm{Tr}(\Sigma^{-1}AA^T) \geq \frac{d}{\sigma_{\min}}\Big(\max ||z - z'||_2^2 + 2||V||_1^2\Big)$ where $\sigma_{\min}$ is the minimum singular value of AA^T. Under our linear filter mechanism, when $\mathbb{E}\Big[D_\alpha\big(\mathcal{N}(\mu + Vy_s + Aw, \Sigma)||\mathcal{N}(\mu, \Sigma)\big)\Big] \leq b$, then such a mechanism satisfies (b, α) Rényi differential privacy.*

Proof of Proposition 2.

$$b \overset{\text{(use assump.)}}{\geq} \mathbb{E}\Big[D_\alpha\Big\{\mathcal{N}(\mu + Vy_s + Aw, \Sigma)||\mathcal{N}(\mu, \Sigma)\Big\}\Big] = \frac{\alpha}{2}\mathbb{E}\Big[(Vy_s + Aw)^T\Sigma^{-1}(Vy_s + Aw)\Big] \tag{3.35a}$$

$$= \frac{\alpha}{2}\mathbb{E}\Big[\ \mathrm{Tr}\left(\Sigma^{-1}(Vy_s + Aw)(Vy_s + Aw)^T\right)\Big] \tag{3.35b}$$

$$= \frac{\alpha}{2}\mathbb{E}\Big[\ \mathrm{Tr}\left(\Sigma^{-1}(Vy_sy_s^TV^T + Vy_sw^TA^T + Awy_s^TV^T + Aww^TA^T)\right)\Big] \tag{3.35c}$$

$$= \frac{\alpha}{2}\ \mathrm{Tr}\left[\mathbb{E}\big(\Sigma^{-1}(Vy_sy_s^TV^T + Aww^TA^T)\big)\right] \tag{3.35d}$$

$$= \frac{\alpha}{2}\ \mathrm{Tr}\left[\mathbb{E}\big(\Sigma^{-1}(Vy_sy_s^TV^T)\big) + \Sigma^{-1}A\mathbb{E}[ww^T]A^T\right] \tag{3.35e}$$

$$= \frac{\alpha}{2}\ \mathrm{Tr}\left[\mathbb{E}\big(\Sigma^{-1}(Vy_sy_s^TV^T)\big) + \Sigma^{-1}AA^T\right] \tag{3.35f}$$

$$= \frac{\alpha}{2}\Big(\mathbb{E}[\mathrm{Tr}\ \underbrace{(y_s^TV^T\Sigma^{-1}Vy_s)}_{>0,\text{as positive definite}}] + \mathrm{Tr}(\Sigma^{-1}AA^T)\Big) \tag{3.35g}$$

$$\geq \frac{\alpha}{2}\ \mathrm{Tr}(\Sigma^{-1}AA^T) \overset{\text{(use assump.)}}{\geq} \frac{\alpha d}{2\sigma_{\min}}(\max ||z - z'||^2 + 2||V||_1^2) \tag{3.35h}$$

$$\overset{\text{(use theorem 3)}}{\geq} D_\alpha(f(z)||f(z')) \tag{3.35i}$$

The $\sigma_{\min}$ is the minimum singular value of AA^T and f is some function mapping.

If Σ is a diagonal matrix, we want to find such an A that holds the following property that

$$b \geq \frac{\alpha}{2}\operatorname{Tr}(\Sigma^{-1}AA^T) \geq \frac{\alpha d}{2\sigma_{(\Sigma,\max)}}\operatorname{Tr}(AA^T) = \frac{\alpha d}{2\sigma_{(\Sigma,\max)}}||A||_F, \tag{3.36}$$

where $\sigma_{(\Sigma,\max)}$ is the max singular value of Σ. Thus, the eigenvalues of AA^T is upper bounded. Therefore, with a prescribed b, a predetermined α, and a known empirical covariance Σ, we can learn a linear filter $\Gamma = [A, V]$. Moreover, if A and V satisfies $\frac{\alpha d}{2\sigma_{(\Sigma,\max)}}||A||_F \leq b$ and $\operatorname{Tr}(\Sigma^{-1}AA^T) \geq \frac{d}{\sigma_{\min}}\big(\max_{z,z'\in Z}||z-z'||^2 + 2||V||_1^2\big)$, then we have (b,α) Rényi differential privacy.

3.6.5 Why does cross-entropy loss work

In this section, we explain the connection between cross-entropy loss and mutual information to give intuition for why maximizing the cross-entropy loss in our optimization reduces the correlation between released data and sensitive labels. Given that the encoder part of the VAE is fixed, we focus on the latent variable z for the remaining discussion in this section.

The mutual information between latent variable z and sensitive label y can be expressed as follows

$$\begin{aligned}
I(z;y) &= \mathbb{E}_{q(z,y)}\Big[\log q(y|z) - \log q(y)\Big] && (3.37)\\
&= \mathbb{E}_{q(z,y)}\big[\log q(y|z) - \log p(y|z) - \log q(y) + \log p(y|z)\big] && (3.38)\\
&= \mathbb{E}_{q(y|z)q(z)}\big[\log \frac{q(y|z)}{p(y|z)}\big] + \mathbb{E}_{q(z,y)}[\log p(y|z)] - \mathbb{E}_{q(y|z)q(y)}[\log q(y)] && (3.39)\\
&= \underbrace{\mathbb{E}_{q(z)}\Big[D_{KL}\big(q(y|z)||p(y|z)\big)\Big]}_{\geq 0} + \mathbb{E}_{q(z,y)}[\log p(y|z)] + H(y) && (3.40)\\
&\geq \mathbb{E}_{q(z,y)}[\log p(y|z)] + H(y), && (3.41)
\end{aligned}$$

where q is the data distribution, and p is the approximated distribution, which is similar to the last logit layer of a neural network classifier. Then, the term

$-\mathbb{E}_{q(z,y)}[\log p(y|z)]$ is the cross-entropy loss $H(q,p)$ (the corresponding negative log-likelihood is $-\mathbb{E}_{q(z,y)}[\log p(y|z)]$). In classification problems, minimizing the cross-entropy loss enlarges the value of $\mathbb{E}_{q(z,y)}[\log p(y|z)]$. Consequently, this pushes the lower bound of $I(z;y)$ in equation (3.41) as high as possible, indicating high mutual information.

However, in our robust optimization, we maximize the cross-entropy, thus, decreasing the value of $\mathbb{E}_{q(z,y)}[\log p(y|z)]$ (more specifically, it is $\mathbb{E}_{q(\tilde{z},y)}[\log p(y|\tilde{z})]$, given the mutual information we care about is between the new representation $\tilde{z}$ and sensitive label y in our application). Thus, the bound of equation (3.41) has a lower value, which indicates the mutual information $I(\tilde{z};y)$ can be lower than before. Such observations can also be supported by the empirical results of mutual information shown in figure 3.9.

3.6.6 Dependency of filter properties and tail bounds

In terms of classifying private attributes, we claim that new perturbed samples $\tilde{z}$ become harder to classify correctly compared to the original samples z. To show this, we further simplify the setting by considering a binary classification case with data $(z,y) \in \mathbb{R}^d \times \{\pm 1\}$. We consider linear classifiers $\theta^T z$ using zero-one loss based on the margin penalty $\xi > 0$. More precisely, we define the loss $\ell_\xi(\theta,(z,y)) = \mathbb{1}\{\theta^T zy \leq \xi\}$. Then the expected loss

$$L_\xi(\theta;z) = \mathbb{E}[\ell_\xi(\theta,(z,y))] = P(\theta^T zy \leq \xi). \tag{3.42}$$

Recall that $\tilde{z} = z + Aw + Vy_s$, where y_s is the one-hot vector that represents y. We have the following expressions:

$$
\begin{aligned}
P(\theta^T zy \leq \xi) &= P\big(\theta^T(\tilde{z} - Aw - Vy_s)y \leq \xi\big) && (3.43)\\
&= P\big(\theta^T \tilde{z}y - y\theta^T Aw - y\theta^T V y_s \leq \xi\big) && (3.44)\\
&\overset{\text{(i)}}{\leq} P\big(\{\theta^T \tilde{z}y \leq \xi\} \cap \big\{ \min\{-\theta^T Aw - \theta^T v_1, \theta^T Aw + \theta^T v_2\} \leq 0\big\}\big) && (3.45)\\
&\overset{\text{(ii)}}{=} P(\theta^T \tilde{z}y \leq \xi) P(\min\{-\theta^T Aw - \theta^T v_1, \theta^T Aw + \theta^T v_2\} \leq 0) && (3.46)\\
&\leq P(\theta^T \tilde{z}y \leq \xi) \max\big\{P(\theta^T Aw \geq -\theta^T v_1), P(\theta^T Aw \leq -\theta^T v_2)\big\} && (3.47)\\
&\overset{\text{(iii)}}{\leq} P(\theta^T \tilde{z}y \leq \xi) \underbrace{\exp\Big(-\frac{||\theta||_2^2 \tau^2}{2(\theta^T AA^T\theta)}\Big)}_{<1}. && (3.48)
\end{aligned}
$$

The inequality (i) uses the fact that the matrix V multiplied by the one-hot vector y_s returns the column vector with index aligned with the non-zero entry in y_s. We explicitly write out column vector v_1, v_2 in this binary classification setting when $y = \pm 1$. The equality (ii) uses the fact that w is the independent noise when we express out y. The inequality (iii) uses the concentration inequality of sub-Gaussian random variables [[125], Lemma 1.3]. Specifically, since $\mathbb{E}(\theta^T Aw) = 0$, we have $P(\theta^T Aw) \leq \exp\big(-\frac{(-\theta^T v_1)^2}{2\theta^T AA^T\theta}\big)$ and $P(\theta^T Aw) \leq \exp\big(-\frac{(\theta^T v_2)^2}{2\theta^T AA^T\theta}\big)$. We then apply the Cauchy-Schwarz inequality on $\theta^T v_1$ (also on $\theta^T v_2$) and let τ be the maximum l_2-norm of v_1 and v_2, i.e. $\tau = \max\{||v_1||_2, ||v_2||_2\}$. Therefore, by rearranging equation (3.48) we show that

$$
P(\theta^T \tilde{z}y \leq \xi) \geq P(\theta^T zy \leq \xi)\Big(\exp\Big(-\frac{||\theta||_2^2 \tau^2}{2(\theta^T AA^T\theta)}\Big)\Big)^{-1} \implies L_\xi(\theta; \tilde{z}) \geq L_\xi(\theta; z), \tag{3.49}
$$

which indicates that classifying $\tilde{z}$ is harder than classifying z under 0-1 loss with the margin based penalty.

Remark: Inequality equation (3.48) provides an interesting insight that a large Frobenius norm of A, i.e. $||A||_{\text{F}}$, gives higher loss on classifying new samples. To see why it holds, we apply the trick $\theta^T AA^T\theta = \text{Tr}(\theta^T AA^T\theta) = \text{Tr}(AA^T\theta\theta^T) = ||A||_{\text{F}}^2 \,\text{Tr}(\theta\theta^T)$.

Thus a large value of $||A||_{\mathrm{F}}^2$ pushes $\exp\left(-\frac{||\theta||_2^2||v^*||_2^2}{2(\theta^T AA^T\theta)}\right)$ to small values, which increases the $L_\xi(\theta;\tilde{z})$.
As mentioned in [115] and [126], other convex decreasing loss functions that capture margin penalty can be surrogates of 0-1 loss, e.g. the hinge loss $L(t) = (1-t)_+$ or logistic loss $L(t) = \log(1+\exp(-t))$ where $t = \theta^T zy$.

3.6.7 Experiment Details

All experiments were performed on Nvidia GTX 1070 8GB GPU with Python 3.7 and Pytorch 1.2.

VAE architecture

The MNIST dataset contains 60000 samples of gray-scale handwritten digits with size 28-by-28 pixels in the training set, and 10000 samples in the testing set. When running experiments on MNIST, we convert the 28-by-28 images into 784 dimensional vectors and construct a network with the following structure for the VAE:

$x \to \mathrm{FC}(300) \to \mathrm{ELU} \to \mathrm{FC}(300) \to \mathrm{ELU} \to \mathrm{FC}(20)(\text{split } \mu \text{ and } \Sigma \text{ to approximate } q(z|x))$
$z \to \mathrm{FC}(300) \to \mathrm{ELU} \to \mathrm{FC}(300) \to \mathrm{ELU} \to \mathrm{FC}(784).$

The UCI-Adult data contains 48842 samples with 14 attributes. Because many attributes are categorical, we convert them into a one-hot encoding version of the input. We train a VAE with the latent dimension of 10 (both mean and variance with dimension of 10 for each) with two FC layers and ELU activation functions for both encoder and decoder.

The UCI-abalone data has 4177 samples with 9 attributes. We pick sex and ring as private and utility labels, leaving 7 attributes to compress down. The VAE is just single linear layer with 4 dimensional output of embedding, having 4-dimensional mean and variance accordingly.

The aligned CelebA dataset contains 202599 samples. We crop each image down to 64-by-64 pixels with 3 color (RGB) channels and pick the first 182000 samples as

the training set and leave the remainder as the testing set. The encoder and decoder architecture for CelebA experiments are described in Table 3.6 and Table 3.7.

Table 3.6: Encoder Architecture in CelebA experiments. We use the DenseNet [127] architecture with a slight modification to embed the raw images into a compact latent vector, with growth rate $m = 12$ and depth $k = 82$

Name	**Configuration**	**Replication**
initial layer	conv2d=(3, 3), stride=(1, 1), padding=(1, 1), channel in = 3, channel out = $2m$	1
dense block1	batch norm, relu, conv2d=(1, 1), stride=(1, 1), batch norm, relu, conv2d=(3, 3), stride=(1, 1), growth rate = m, channel in = $2m$	12
transition block1	batch norm, relu, conv2d=(1, 1), stride=(1, 1), average pooling=(2, 2), channel in = $\frac{(k-4)}{6}m + 2m$, channel out = $\frac{(k-4)}{12}m + m$	1
dense block2	batch norm, relu, conv2d=(1, 1), stride=(1, 1), batch norm, relu, conv2d=(3, 3), stride=(1, 1), growth rate=m, channel in = $\frac{(k-4)}{12}m + m$,	12
transition block2	batch norm, relu, conv2d=(1, 1), stride=(1, 1), average pooling=(2, 2), channel in = $\frac{(k-4)}{6}m + \frac{(k-4)}{12}m + m$ channel out= $\frac{1}{2}\big(\frac{(k-4)}{6}m + \frac{(k-4)}{12}m + m\big)$	1
dense block3	batch norm, relu, conv2d=(1, 1), stride=(1, 1), batch norm, relu, conv2d=(3, 3), stride=(1, 1), growth rate = m, channel in = $\frac{1}{2}\big(\frac{(k-4)}{6}m + \frac{(k-4)}{12}m + m\big)$	12
transition block3	batch norm, relu, conv2d=(1, 1), stride=(1, 1), average pooling=(2, 2), channel in = $\frac{1}{2}\big(\frac{(k-4)}{6}m + \frac{(k-4)}{12}m + m\big) + \frac{(k-4)}{6}m$ channel out = $\frac{1}{2}\big(\frac{1}{2}\big(\frac{(k-4)}{6}m + \frac{(k-4)}{12}m + m\big) + \frac{(k-4)}{6}m\big)$	1
output layer	batch norm, fully connected 100	1

Filter Architecture

We use a generative linear filter throughout our experiments. In the MNIST experiments, we compressed the latent embedding down to a 10-dim vector. For **MNIST**

Table 3.7: Decoder Architecture in CelebA experiments.

Name	Configuration	Replication
initial layer	fully connected 4096	1
reshape block	resize 4096 to $256 \times 4 \times 4$	1
deccode block	conv transpose=(3, 3), stride=(2, 2), padding=(1, 1), outpadding=(1, 1), relu, batch norm	4
decoder block	conv transpose=(5, 5), stride=(1, 1), padding=(2, 2)	1

Case 1, we use a 10-dim Gaussian random vector w concatenated with a 10-dim one-hot vector y_s representing digit id labels, where $w \sim \mathcal{N}(0, I)$ and $y_s \in \{0, 1\}^{10}$. We use the linear filter Γ to ingest the concatenated vector and add the corresponding output to the original embedding vector z to yield $\tilde{z}$. Thus the mechanism is

$$\tilde{z} = f(z, w, y) = z + \Gamma \begin{bmatrix} w \\ y_s \end{bmatrix}, \tag{3.50}$$

where $\Gamma \in \mathbb{R}^{10\times 20}$ is a matrix. For **MNIST Case 2**, we use a similar procedure except the private label y is a binary label (i.e. digit value ≥ 5 or not). Thus, the corresponding one-hot vector is 2-dimensional. Since we keep w to be a 10-dimensional vector, the corresponding linear filter Γ is a matrix in $\mathbb{R}^{10\times 12}$.

In the experiment of CelebA, we create the generative filter following the same methodology in equation (3.50), with some changes on the dimensions of w and Γ because images of CelebA are bigger than MNIST digits.[4] Specifically, we set $w \in R^{50}$ and $A \in \mathbb{R}^{50\times 52}$.

Adversarial classifiers

In the MNIST experiments, we use a small architecture consisting of neural networks with two fully connected layers and an exponential linear unit (ELU) to serve as the

[4]We use a VAE type architecture to compress the image down to a 100 dimensional vector, then enforce the first 50 dimensions as the mean and the second 50 dimensions as the variance

privacy classifiers, respectively. The specific structure of the classifier is depicted as follows:

$$z \text{ or } \tilde{z} \to \text{FC}(15) \to \text{ELU} \to \text{FC}(y)[\text{ or FC}(u)].$$

We use linear classifier for UCI-adults and UCI-abalone with input dimension of 10 and 4 which aligns with the embedding dimensions respectively. In the CelebA experiments, we construct a two-layered neural network as follows:

$$z \text{ or } \tilde{z} \to \text{FC}(60) \to \text{ELU} \to \text{FC}(y)[\text{ or FC}(u)].$$

The classifiers ingest the embedded vectors and output unnormalized logits for the private label or utility label. The classification results of CelebA can be found in Table 3.4.

Other hyper-parameters

When we first train the VAE type models using loss function in equation (3.15), we pick multiple values such as $\gamma = \{0.01, 0.1, 1, 10, 100\}$, $\kappa = \{1, 10\}$, and $C_L = \{2, 4, 6\}$ to evaluate the performance. A combination of $\gamma = 0.1, \kappa = 1, C_L = 4$ yields the smallest loss among all the options.

In the min-max training using the objective in equation (3.16), we pick multiple betas ($\beta = \{0.1, 0.5, 1, 2, 4\}$) and report the results when $\beta = 2$ (in MNIST and CelebA) and $\beta = 1$ (in UCI-Adult and UCI-abalone) because this gives the largest margin between accuracy of utility label and accuracy of private label [e.g. highest $(acc_u - acc_y)$]. We also used the relaxed soft constraints mentioned in equation (3.18) by setting $\lambda_1 = \lambda_2 = 1000$ and divide them in halves every 500 epochs with a minimum clip value of 2. We train 1000 epochs for MNIST, UCI-Adult, and UCI-abalone, and 10000 epochs for CelebA.

We use Adam optimizer [128] throughout the experiments with learning rate 0.001 and batch size 128 for MNIST, UCI-Adult and UCI-abalone. In CelebA experiment, the learning rate is 0.0002 and the batch size is 24.

Distributed training setting

This section provides a more detailed look into how our scheme can be run in a local and distributed fashion through an example experiment on the MNIST dataset with 2 independent users. The first user adopts the label of digit ≥ 5 or not as private and odd or even as the utility label. The second user prefers the opposite and wishes to privatize odd or even and maintain ≥ 5 or not as the utility label. We first partition the MNIST dataset into 10 equal parts where the first part belongs to one user and the second part belongs to the other user. The final eight parts have already been made suitable for public use either through privatization or because they do not contain information that their original owners have considered sensitive. Each part is then encoded into their 10-dimensional representations and passed onto the two users for the purpose of training an appropriate classifier rather than one trained on a single user's biased dataset. Since the data is already encoded into its representation, the footprint is very small when training the classifiers. Then, the generative filter for each user is trained separately and only on the single partition of personal data. Meanwhile, the adversarial and utility classifiers for each user are trained separately and only on the 8 parts of public data combined with the one part of personal data. The final result is 2 generative filters, one for each user, that correspond to their own choice of private and utility labels. After privatization through use of the user specific filters, we can evaluate the classification accuracy on the private and utility labels as measured by adversaries trained on the full privatized dataset, which is the combination of each users privatized data.

3.6.8 More results of MNIST experiments

In this section, we illustrate detailed results for the MNIST experiment when we set whether the digit is odd or even as the utility label and whether the digit is greater than or equal to 5 as the private label. We first show samples of raw images and privatized images in Figure 3.6. We show the classification accuracy and its sensitivity in Figure 3.7. Furthermore, we display the geometry of the latent space in Figure 3.8.

In addition to the classification accuracy, we evaluate the mutual information, to confirm that our generative filter indeed decreases the correlation between released data and private labels, as shown in Figure 3.9.

Utility of Odd or Even

We present some examples of digits when the utility is an odd or even number (Figure 3.6). The confusion matrix in Figure 3.7a shows that false positive rate and false negative rate are almost equivalent, indicating the perturbation resulting from the filter doesn't necessarily favor one type (pure positive or negative) of samples. Figure 3.7b shows that the generative filter, learned through minmax robust optimization, outperforms the Gaussian mechanism under the same distortion budget. The Gaussian mechanism reduces the accuracy of both private and utility labels, whereas the generative filter can maintain the accuracy of the utility while decreasing the accuracy of the private label, as the distortion budget goes up.

Furthermore, the distortion budget prevents the generative filter from distorting non-target attributes too severely. This budget allows the data to retain some information even if it is not specified in the filter's loss function. Figure 3.7c shows the classification accuracy with the added non-target label of circle from MNIST case 1.

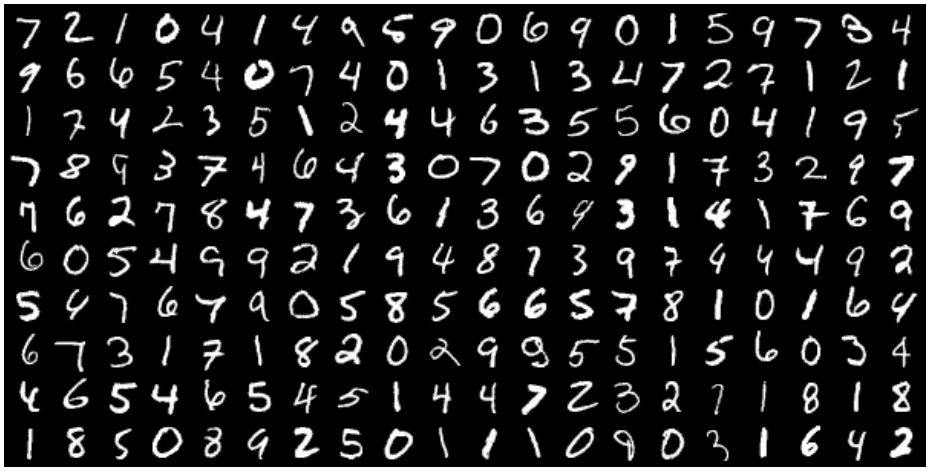
Empirical Mutual information

We use the empirical mutual information [129] to verify if our new perturbed data is less correlated with the sensitive labels from an information-theoretic perspective. The empirical mutual information is clearly decreased as shown in Figure 3.9, a finding that supports our generative adversarial filter can protect the private label given a certain distortion budget.

3.6.9 Additional information on CelebA experiments

Comments on comparison of classification accuracy

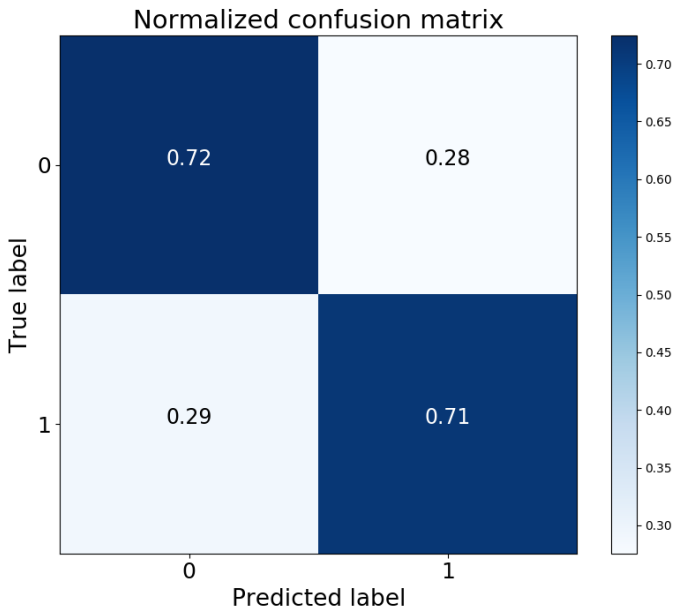
We notice that our classification results on utility label (smiling) in the CelebA experiment perform worse than the state of the art classifiers presented in [106] and



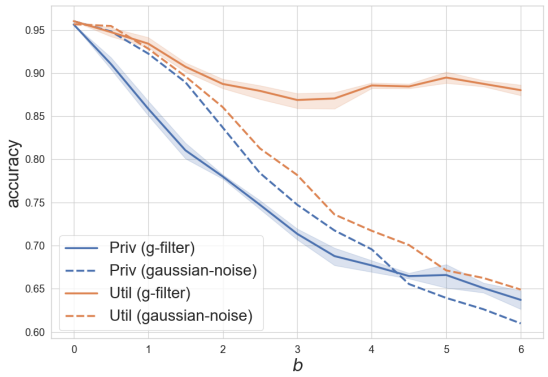
(a) Sample of original digits

(b) Same images large-valued digits privatized

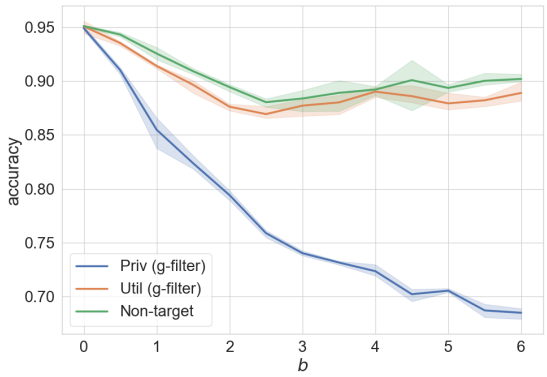
Figure 3.6: **MNIST case 2**: Visualization of digits pre- and post-noise injection and adversarial training. We discover that some large-valued digits (≥ 5) are randomly switched to low-valued (< 5) digits (or vice versa) while some even digits remain even digits and odd digits remain as odd digits.



(a) Confusion matrix of classifying the private label when $b = 3$



(b) Sensitivity of classification accuracy versus the distortion b



(c) Sensitivity of classification accuracy versus the distortion b with a non-target attribute

Figure 3.7: **MNIST case 2**: Figure 3.7a shows the false positive and false negative rates for classifying the private label when the distortion budget is 3 in KL-divergence. The Figure 3.7b shows that when we use the generative adversarial filter, the classification accuracy of private labels drops from 95% to almost 65% as the distortion increases, while the utility label can still maintain close to 90% accuracy throughout. Meanwhile, the additive Gaussian noise performs worse because it yields higher accuracy on the private label and lower accuracy on the utility label, compared to the generative adversarial filter. Figure 3.7c shows how non-target attributes not included in the filter's loss function (circle) can still be preserved due to the distortion budget restricting noise injection. The error bars show the standard error over a batch of 10 samples

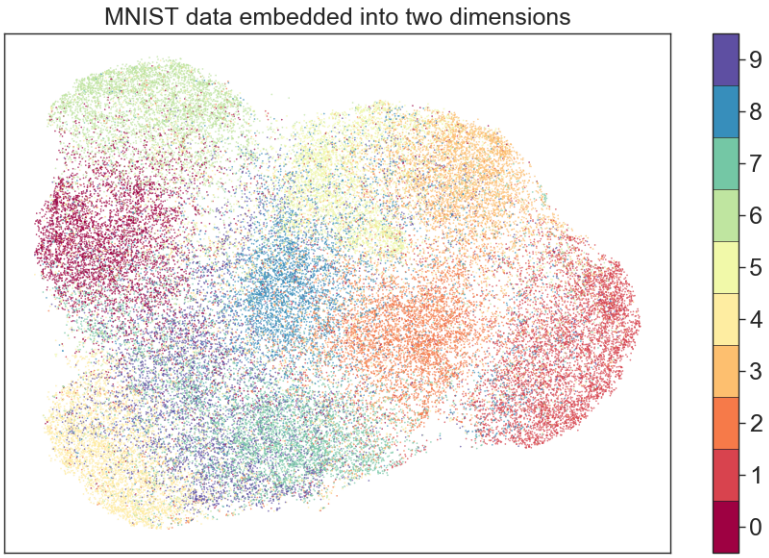
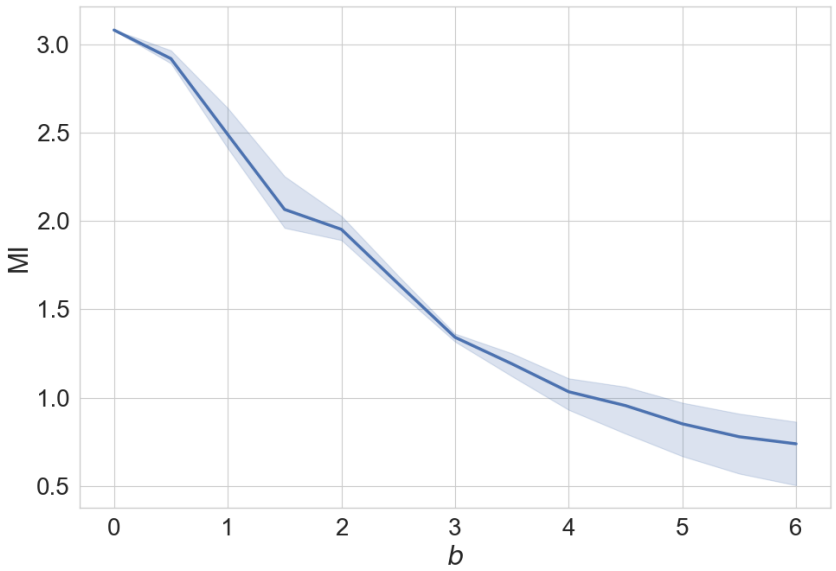
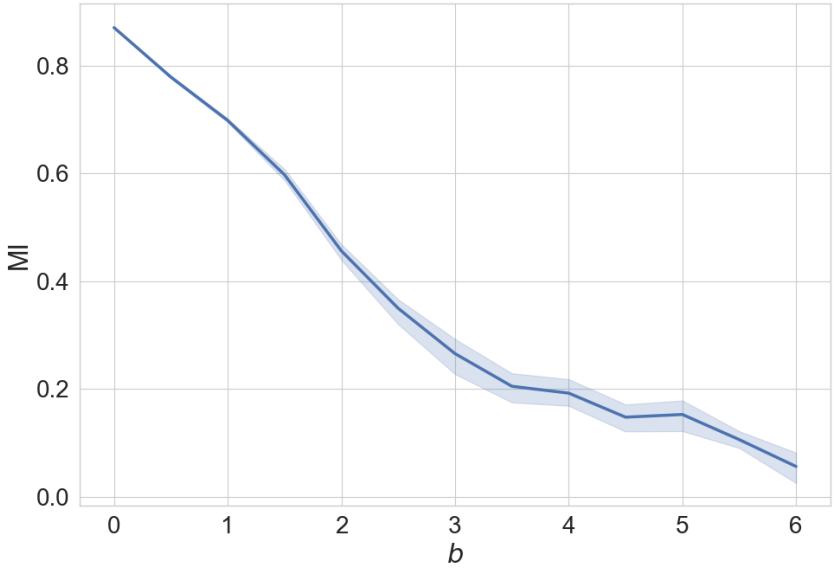


Figure 3.8: **MNIST case 2**: Visualization of the perturbed latent geometry. We discover that 0 is closer to 6, compared with the original latent geometry in Figure 3.4a, a finding that indicates that it would be more difficult to distinguish which of those two digits is larger than or equal to five, even though both are even digits.

[110]. However, the main purpose of our approach is not building the best image classifier, but getting a baseline of a comparable performance on original data without the noise perturbation. Instead of constructing a large feature vector (through convolution, pooling, and non-linear activation operations), we compress a facial image down to a 50-dimension vector as the embedding. We make the perturbation through a generative filter to yield a vector with the same dimensions. Finally, we construct a neural network with two fully-connected layers and an elu activation after the first layer to perform the classification task. We believe the deficit of the accuracy is because of the compact dimensionality of the representations and the simplified structure of the classifiers. We expect that a more powerful state of the art classifier trained on the released private images will still demonstrate the decreased accuracy on the private labels compared to the original non-private images while maintaining higher accuracy on the utility labels. This hypothesis is supported by the empirically measured decrease in mutual information demonstrated in section 3.6.8.



(a) Digit identity as the private label



(b) Large- or small-value as the private label

Figure 3.9: Mutual information between the perturbed embedding and the private label decreases as the distortion budget grows, for both MNIST case 1 and case 2.

More examples of CelebA

In this part, we illustrate more examples of CelebA faces yielded by our generative adversarial filter (Figures 3.11, 3.12, and 3.13). We show realistic looking faces generated to privatize the following labels: attractive, eyeglasses, and wavy hair, while maintaining smiling as the utility label. The blurriness of the images is typical of state of the art VAE models because of the compactness of the latent representation [130, 131]. The blurriness is not caused by the privatization procedure but by the encoding and decoding steps as demonstrated in Figure 3.10.

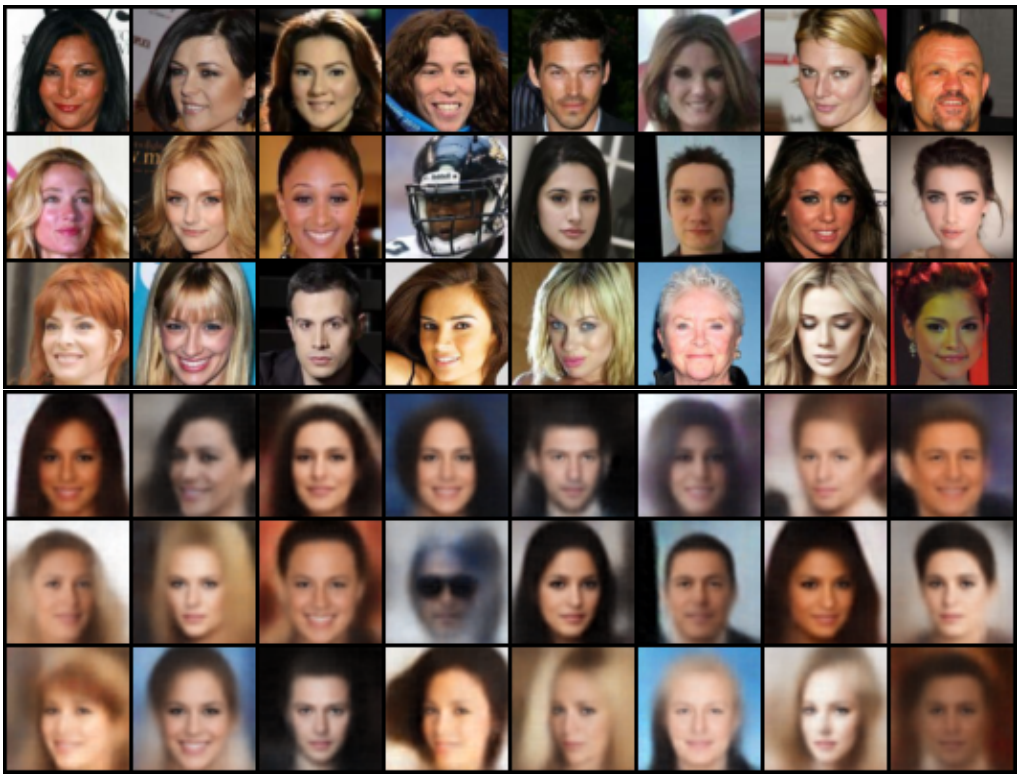
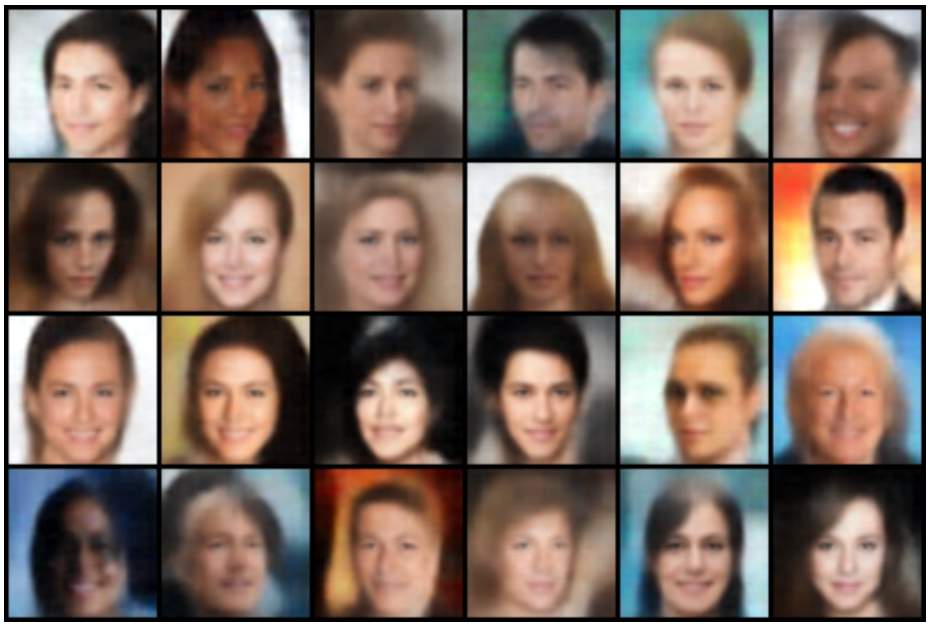


Figure 3.10: Visualizing raw samples (on left) and encoded-decoded samples (on right) from a trained VAE with the Lipschitz smoothness.

3.6.10 Deferred equations

In this section, we describe the f-divergence between two Gaussian distributions. The characterization of such a divergence is often used in previous derivations.

(a) raw



(b) attractive privatized

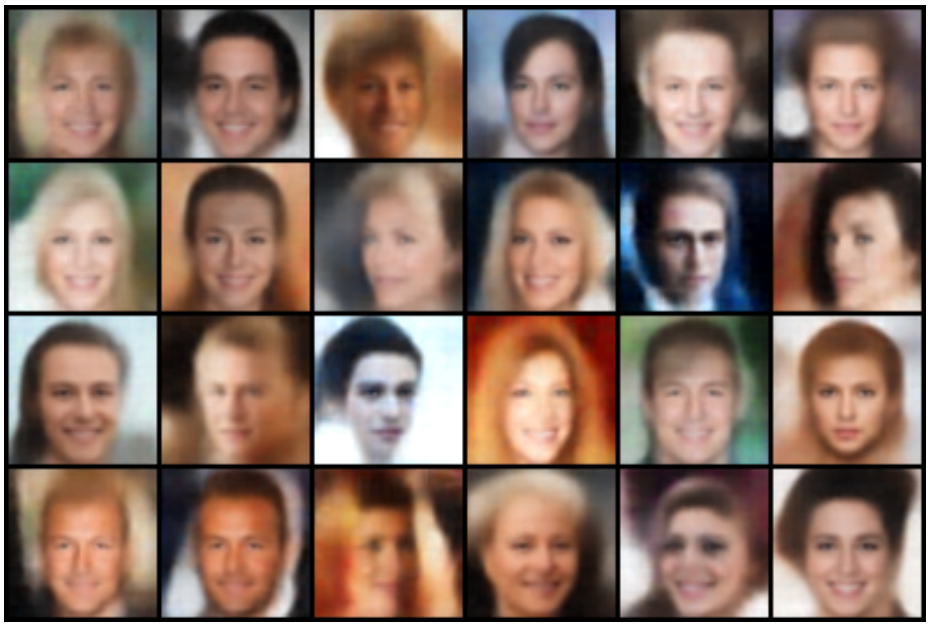
Figure 3.11: Sampled images. We find some non-attractive faces switch to attractive faces and some attractive looking images are changed into non-attractive, from Figure 3.11a to Figure 3.11b.

(a) raw

(b) eyeglass privatized

Figure 3.12: Sampled images. We find some faces with eyeglasses are switched to non-eyeglasses faces and some non-eyeglasses faces are changed into eyeglasses-wearing faces, from figure 3.12a to figure 3.12b.

(a) raw



(b) wavy hair privatized

Figure 3.13: Sampled images. We discover that some faces with wavy hair switch to images with non-wavy hair after our filter's perturbation (and vice versa), from figure 3.13a to figure 3.13b.

Rényi divergence between Gaussian distributions

α-Rényi divergence between two multivariate Gaussians [[124] and [132], Appendix Proposition 6]:

$$D_\alpha(p_i||p_j) = \frac{\alpha}{2}(\mu_i - \mu_j)^T\Sigma_\alpha^{-1}(\mu_i - \mu_j) - \frac{1}{2(\alpha-1)}\log\Big(\frac{|\Sigma_\alpha|}{|\Sigma_i|^{1-\alpha}|\Sigma_j|^\alpha}\Big), \tag{3.51}$$

where $\Sigma_\alpha = \alpha\Sigma_j + (1-\alpha)\Sigma_i$ and $\alpha\Sigma_i^{-1} + (1-\alpha)\Sigma_j^{-1} > 0$ is positive definite.
We give a specific example of two Gaussian distributions p and q that are $\mathcal{N}(\mu_1, \Sigma)$ and $\mathcal{N}(\mu_2, \Sigma)$ respectively. Letting $\mathbb{E}_{\mu_2}$ denote expectation over $x \sim \mathcal{N}(\mu_2, \Sigma)$, we then have

$$D_\alpha(p||q) = \frac{1}{\alpha-1}\log\int\Big(\frac{p(x)}{q(x)}\Big)^\alpha q(x)dx \tag{3.52}$$

$$= \frac{1}{\alpha-1}\log\mathbb{E}_{\mu_2}\Big[\exp\Big(-\frac{\alpha}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{\alpha}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\Big)\Big] \tag{3.53}$$

$$\overset{\text{(i)}}{=} \frac{1}{\alpha-1}\log\mathbb{E}_{\mu_2}\Big[\exp\Big(-\frac{\alpha}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \alpha(\mu_1-\mu_2)^T\Sigma^{-1}(x-\mu_2)\Big)\Big] \tag{3.54}$$

$$\overset{\text{(ii)}}{=} \frac{1}{\alpha-1}\log\Big(\exp\Big[\frac{-\alpha}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \frac{\alpha^2}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)\Big]\Big) \tag{3.55}$$

$$= \frac{1}{\alpha-1}\frac{(\alpha-1)\alpha}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) \tag{3.56}$$

$$= \frac{\alpha}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2), \tag{3.57}$$

where equality (i) uses the relationship $-(x-a)^2 + (x-b)^2 = -(a-b)^2 + 2(a-b)(x-b)$ and equality (ii) uses a linear transformation of Gaussian random variables $(\mu_1-\mu_2)^T\Sigma^{-1}(x-\mu_2) \sim \mathcal{N}\big(0, (\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)\big)$.

Chapter 4

Energy Resource Control via Privacy Preserving Data

Although the frequent monitoring of smart meters enables granular control over energy resources, it also increases the risk of leakage of private information such as income, home occupancy, and power consumption behavior that can be inferred from the data by an adversary. We propose a method of releasing modified smart meter data so specific private attributes are obscured while the utility of the data for use in an energy resource controller is preserved. The method privatizes data by injecting noise conditioned on the private attribute through a linear filter learned via a minimax optimization. The optimization contains the loss function of a classifier for the private attribute, which we maximize, and the energy resource controller's objective formulated as a canonical form optimization, which we minimize. We perform our experiment on an aggregated dataset of household consumption with solar generation and another from the Commission for Energy Regulation (CER) that contains household smart meter data with sensitive attributes such as income and home occupancy. We demonstrate on the CER data that our method is able to reduce the ability of an adversary to classify a binary income label to that of random guessing while maintaining an objective value for an energy storage controller within 10% of optimal.

4.1 Introduction

Traditionally, the power grid has been managed by the producers and grid operators with information primarily exchanged among the large asset owners with little feedback from its end users. However, the push for renewable energy sources has brought about the rise of distributed energy resources (DERs) that lie under the control of many smaller and disparate users, causing a paradigm shift in the flow of information. The successful operation of DERs and other smart grid technologies depends on the exchange of large amounts of data from many different end users [133, 134, 135]. Due to increased regulations [136], it may be unrealistic to assume data will be available without consideration of the data owners' privacy. The increased granularity of data required for smart grid operation enables the inference of personal information [137] such as household income, which suggests data owners may be reluctant to exchange their data without some effort towards preserving privacy.

Many studies have investigated approaches to protect smart meter data privacy using a number of different techniques and metrics with detailed surveys given in [138, 139, 140]. The recent paper from Giaconi et al. [140] defines two general types of approaches, user demand shaping, and data manipulation, with the latter broken into further categories such as data obfuscation or aggregation.

Some papers in the data manipulation category, [141] and [142], perform a pre-processing step on the raw data in order to better prepare it for its end use; however, the pre-processing only considers conditioning the data for its utility without explicitly defining the objective of preserving privacy. Therefore, the pre-processing step may be insufficient to prevent sensitive information from being inferred from the processed data. On the other hand, the aggregation technique presented in [143] and [144], provides user privacy by aggregating data until the aggregate does not reflect on any specific meter data. However, the aggregation group size can be on the order of thousands and there is no consideration to the cost of data utility as a result of aggregation.The data obfuscation category of approaches often come with similar limitations. For example, many studies come from differential privacy (DP) [145], which is widely adopted in designing and analyzing privacy mechanisms in the context of energy data [146, 147, 148, 149, 150, 151]. Specifically, studies [146, 147, 148]

proposed several frameworks for reducing the mutual information between raw data and privatized data (e.g. power profiles), [149] investigated the differential privacy effect with some noise injection (e.g. Laplace noise), and [150] explored how much noise must be added to the data in order to achieve a certain level of differential privacy for an existing Laplace mechanism in the context of solving optimal power flow. Similarly to the aggregation approach and opposite to the pre-processing approaches, these DP approaches typically only consider obfuscating the data for privacy without simultaneously considering the utility of the data. Therefore, after achieving privacy, the data may be too obfuscated to be useful. One paper that avoids this issue is [151], which proposes a DP mechanism to release the state parameters of power networks with a guarantee of the feasibility of the alternating current (AC) power flow problem. By guaranteeing AC-feasibility of their data, they are making a step in ensuring the data still retains utility after privatization.

The user demand shaping category of approaches often involves a balance between utility and privacy since the privacy is achieved via device operation as opposed to data manipulation [140]; however, achieving privacy through device operation comes with limitations such as its efficacy depends on the physical capabilities of the devices.

We distinguish our studies by developing a methodology that learns an optimal noise injection on the data that balances the trade off between privacy and data utility, thus, preserving as much utility in the data as possible. Our method falls within the data obfuscation category, but differs from strict differential privacy [145] because we use a general notion of privacy that reduces the correlation between private attributes and the data. This general notion of privacy gives us the flexibility to maintain the utility of the data while still eliminating an adversary's ability to recognize certain private attributes. Since many applications of smart meter data involve their use in optimization procedures, we define the utility as the performance achieved when such data is used for optimal control [152]. We consider a scenario where individual owners of DERs, such as battery storage systems, wish to privatize their data before releasing it to a DER aggregator to make optimal control decisions on their behalf, which can have applications in the context of [133], [134], and [135],. This scenario makes our approach share some similarity to the user demand shaping category of privatization

methods in that we provide balance between the utility of DERs and privacy; however, it differs in that our privatization occurs on the data before the operation of the DERs rather than on the actual power consumption after the operation of DERs.

Our work contributes to the research of smart meter privacy in following ways. We propose a minimax approach to generate realistic meter data that is decorrelated from sensitive attributes while maintaining limited performance loss of a cost minimization optimal control algorithm using battery storage. Additionally, we developed a parallelized method that can be easily incorporated in modern deep learning architectures. The correlation of data privatized by our method with sensitive attributes and the performance of a control algorithm is evaluated on two real datasets of residential power demand: one with synthetic sensitive labels and one with real labels. We demonstrate that our method is able to decrease the classification accuracy of an adversary by over 20% while maintaining the performance of the optimization to within 10% over both datasets.

The rest of the paper is organized as follows: we describe the energy resource control in section 4.2, control with privatized data generated from the minimax learning algorithm in section 4.3, experiments and results on the two datasets in section 4.4, and the Conclusion in Section section 4.5.

4.2 Energy Resource Control

4.2.1 Notation

We use bold letters for vectors and matrices and regular letters for scalars. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \boldsymbol{y}$ represents the element-wise order $\boldsymbol{x}(i) \geq \boldsymbol{y}(i)$ for $i \in [n]$ where $[n]$ denotes the set $[n] = \{1, \ldots, n\}$. And $\boldsymbol{x} \geq 0$ means all elements in the vector are not less than the scalar zero. We make the dependence on the underlying probability distribution P when we write expectations (e.g. $\mathbb{E}_P[X]$ where X denotes a random variable). The Frobenius norm of a matrix $\boldsymbol{A}$ is $||\boldsymbol{A}||_F$. We write $\nabla_\theta \mathcal{L}(\theta; X)$ or $\mathsf{d}\mathcal{L}(\theta; X)$, where we typically mean differentiation of the loss function $\mathcal{L}$ with respect to the parameter $\theta \in \mathbb{R}^n$. $\mathcal{N}$ stands for Normal (or Gaussian) distribution and $\mathbb{R}_+$ denotes the non-negative real numbers. We use $:=$ to represent "define as."

All the vectors are column vectors by default unless we explicitly address otherwise in a specific context.

4.2.2 Battery storage control

Control with deterministic demand: Consider a basic battery control problem with the goal of minimizing the energy cost given a prescribed price $\boldsymbol{p} \in \mathbb{R}^H$, where H is the time horizon, typically 24 for an hourly price. An uncontrollable electricity demand is specified as $\boldsymbol{d} \in \mathbb{R}_+^H$. We denote the decision variables for battery control to be $\boldsymbol{x}$ and expand it into $\boldsymbol{x}_{in}, \boldsymbol{x}_{out}, \boldsymbol{x}_s \in \mathbb{R}_+^H$ each of which represents the charging, discharging, and the amount of charge in storage, i.e. $\boldsymbol{x}^\intercal = [\boldsymbol{x}_{in}^\intercal, \boldsymbol{x}_{out}^\intercal, \boldsymbol{x}_s^\intercal]$. The battery optimal control is formulated as follows (**Problem4.1**):

$$\begin{aligned}
\min_{\boldsymbol{x}} \; & \boldsymbol{p}^\intercal(\boldsymbol{x}_{in} - \boldsymbol{x}_{out} + \boldsymbol{d})_+ + \beta_1 ||\boldsymbol{x}_{in}||_2^2 + \beta_2 ||\boldsymbol{x}_{out}||_2^2 \\
& + \beta_3 ||\boldsymbol{x}_s - \alpha B||_2^2 && \text{(4.1a)} \\
\text{s.t.} \quad & \boldsymbol{x}_s(j+1) = \boldsymbol{x}_s(j) - \frac{1}{\eta_{out}} \boldsymbol{x}_{out}(j) + \eta_{in} \boldsymbol{x}_{in}(j) \quad \forall j \in [H] && \text{(4.1b)} \\
& \boldsymbol{x}_s(1) = B_{init} && \text{(4.1c)} \\
& 0 \leq \boldsymbol{x}_{in} \leq c_{in} && \text{(4.1d)} \\
& 0 \leq \boldsymbol{x}_{out} \leq c_{out} && \text{(4.1e)} \\
& 0 \leq \boldsymbol{x}_s \leq B. && \text{(4.1f)}
\end{aligned}$$

The linear term (with respect to $\boldsymbol{x}$) in the objective is the cost of electricity when there is no value for selling the energy back to the grid. This condition represents a situation where there are no net-metering incentives. The quadratic penalty terms $\beta_1 ||\boldsymbol{x}_{in}||_2^2$ and $\beta_2 ||\boldsymbol{x}_{out}||_2^2$ are added to protect the battery state of health in the horizon [153]. The term $\beta_3 ||\boldsymbol{x}_s - \alpha B||_2^2$ is added to set the battery state to be close to the target value αB with B as the battery size and $\alpha \in (0, 1)$. $\beta_1, \beta_2, \beta_3$ are hyper-parameters to control these penalties. c_{in} and c_{out} are the charging-in and discharging-out power capacities. And the parameter η_{in} and η_{out} denote the charging and discharging efficiency (between 0 and 1). The constraint (4.1b) indicates that the battery state in the next timestep equals the current battery state adding up the net charging

amount (summing up charging and discharging together). Constraint (4.1c) sets the initial state of the battery to be B_{init}. To simplify the notation, we define a set $\mathcal{X} := \{\boldsymbol{x}|(4.1b)$ - $(4.1f)$ are feasible for some $\boldsymbol{x} \in \mathbb{R}^{3H}\}$. Hence, we use $\boldsymbol{x} \in \mathcal{X}$ to succinctly express that $\boldsymbol{x}$ satisfies the battery constraints. We convert the problem (4.1) into a canonical convex form in Appendix 4.6.2 and develop a paralleled algorithm that makes use of automatic differentiation, open-source convex solvers, and pytorch[154]– a popular deep learning framework.

Control with stochastic demand: When determining the control with an uncertain demand, we minimize the expected cost under some demand distribution P. The objective is slightly changed as follows (**Problem4.2**):

$$\begin{aligned} \min \mathcal{L}_u(\boldsymbol{x}, \boldsymbol{d}) := \min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{d}\sim P}\left[\boldsymbol{p}^\intercal(\boldsymbol{x}_{in} - \boldsymbol{x}_{out} + \boldsymbol{d})_+\right] \\ + \beta_1||\boldsymbol{x}_{in}||_2^2 + \beta_2||\boldsymbol{x}_{out}||_2^2 + \beta_3||\boldsymbol{x}_s - \alpha B||_2^2 \end{aligned} \tag{4.2a}$$

$$\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X}. \tag{4.2b}$$

Since there is uncertainty behind what the privatized demand will be during training, we use the formulation of the stochastic problem to motivate the minimax problem used for training in section 4.3.2. The details behind the training methodology is presented in the following section.

4.3 Control with Privatized Demand

Protecting privacy in our context means reducing the correlation between the smart meter data and the sensitive attribute of the data owner, e.g. income or square-footage of the house. We justify why such a consideration of privacy protection is useful in practice in section 4.3.1.

4.3.1 Revealing privacy from data

In this section, we consider a simple scenario in which the sensitive information is a binary label, such as a small or large home, which can be inferred from smart meter data. Given the raw demand $\mathbf{d} \in \mathbb{R}_+^H$ and sensitive label y $\in \{0, 1\}$, the adversary

builds a classifier f_ψ that takes in demand $\boldsymbol{d}$ to estimate y with a prescribed loss function $\mathcal{L}_a$. Specifically, we assume the adversary minimizes the classification loss

$$\min_{\psi} \mathcal{L}_a\big(f_\psi(\boldsymbol{d}), \mathrm{y}\big)$$

to infer the private information y. A popular choice of classification loss is cross-entropy loss (or log-loss)[155]. That is

$$\min_{\psi} \Big\{ -\mathrm{y}\log(f_\psi(\boldsymbol{d})) - (1-\mathrm{y})\log\big(1 - f_\psi(\boldsymbol{d})\big) \Big\}$$

when y is a binary variable. The classifier f_ψ is parameterized by ψ and can be a neural network that outputs an estimate of the probability of the positive label. Previous studies [6, 14] showed that estimating a sensitive label such as income or square-footage of the house reaches 69% accuracy using features of smart meter data and models like the support vector machine or random forest. We use an alternative neural network model that leverages the daily power consumption (demand) and achieves state-of-the-art accuracy of the private label. More details can be found in section 4.4.

4.3.2 Control with private demand

Our goal is to minimize the cost of energy while incorporating privacy protection. Specifically, we design a *data generator* that creates a perturbed version of the raw demand data in a way that increases the adversarial classification loss, while enabling an optimal controller to minimize the energy cost. From a modeling perspective, we have a minimax problem (**Problem4.3**):

$$\min_{\boldsymbol{G}} \mathcal{L}_u\big(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}), \boldsymbol{d}\big) + \lambda_a \max_{\boldsymbol{G}} \min_{\psi} \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y}) \tag{4.3a}$$

$$\text{s.t. } \tilde{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{G}\begin{bmatrix}\boldsymbol{\varepsilon}\\ \boldsymbol{y}\end{bmatrix}, \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{4.3b}$$

$$\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}) = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}_u(\boldsymbol{x}, \tilde{\boldsymbol{d}}), \tag{4.3c}$$

where the parameter $\boldsymbol{G}$ is a matrix that affects the distribution of $\tilde{\boldsymbol{d}}$. In this case, we consider a linear transformation of Gaussian noise $\boldsymbol{\varepsilon}$. Variable $\boldsymbol{y}$ is the one-hot encoding of the sensitive binary label, and f_ψ is a classifier that takes in the perturbed demand data and predicts the corresponding label. The $\mathcal{L}_u$ stands for utility loss. It is important to note that $\mathcal{L}_u$ in the objective uses the raw demand to evaluate the cost of the control decisions determined using the perturbed demand. This represents the case where the storage unit acts on the perturbed information, but the real world value is based on the original raw data.

In order to solve the non-trivial optimization (4.3), we simplify the constraints (further explained in section 4.3.3)) and make use of adversarial training, which is a common technique in studies of generative adversarial networks (GAN) and their applications [102, 156]. We add a regularization term $\mathbb{E}||\tilde{\boldsymbol{d}}-\boldsymbol{d}||_2^2$ in the objective with an additional hyper-parameter κ,

$$\min_{\boldsymbol{G}} \mathcal{L}_u\big(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}), \boldsymbol{d}\big) - \lambda_a \mathcal{L}_a(f(\tilde{\boldsymbol{d}}), \boldsymbol{y}) + \kappa \mathbb{E}||\tilde{\boldsymbol{d}} - \boldsymbol{d}||_2^2, \tag{4.4}$$

which helps convergence of the training and preserves parts of the demand that are not related to the privacy or utility loss instead of allowing them to be perturbed arbitrarily.

We denote matrix $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$ with $\Gamma \in \mathbb{R}^{H\times H}$ and $\boldsymbol{V} \in \mathbb{R}^{H\times 2}$. The altered demand then becomes $\tilde{\boldsymbol{d}} = \boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}y$. By denoting π to be the prior distribution of one-hot labels, e.g. $\pi = [p, 1-p]^\intercal$ where p is the prior probability of a positive

label, we can rewrite the distortion regularization as

$$
\begin{aligned}
&\mathbb{E}(||\tilde{\boldsymbol{d}} - \boldsymbol{d}||_2^2) = \mathbb{E}\big[||\boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y} - \boldsymbol{d}||_2^2\big] \\
&= \mathbb{E}\big[(\Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y})^\intercal(\Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y})\big] \\
&= \mathbb{E}(\boldsymbol{\varepsilon}^\intercal\Gamma^\intercal\Gamma\boldsymbol{\varepsilon} + \boldsymbol{y}^\intercal\boldsymbol{V}^\intercal\boldsymbol{V}\boldsymbol{y} + \boldsymbol{y}^\intercal\boldsymbol{V}^\intercal\Gamma\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\intercal\Gamma^\intercal\boldsymbol{V}\boldsymbol{y}) \\
&\overset{\text{(i)}}{=} \mathbb{E}\big[\operatorname{Tr}(\Gamma\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\intercal\Gamma^\intercal) + \operatorname{Tr}(\boldsymbol{V}\boldsymbol{y}\boldsymbol{y}^\intercal\boldsymbol{V}^\intercal)\big] \\
&\overset{\text{(ii)}}{=} \operatorname{Tr}(\Gamma\underbrace{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\intercal]}_{\boldsymbol{I}}\Gamma^T) + \operatorname{Tr}\Big(\begin{bmatrix} | & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 \\ | & | \end{bmatrix}\underbrace{\begin{bmatrix} p^2 & p(1-p) \\ p(1-p) & (1-p)^2 \end{bmatrix}}_{\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\intercal]}\begin{bmatrix} - & \boldsymbol{v}_1^\intercal & - \\ - & \boldsymbol{v}_2^\intercal & - \end{bmatrix}\Big) \\
&\overset{\text{(iii)}}{=} \operatorname{Tr}(\Gamma\Gamma^\intercal) + ||p\boldsymbol{v}_1 + (1-p)\boldsymbol{v}_2||_2^2 \\
&= ||\Gamma||_F^2 + ||\boldsymbol{V}\pi||_2^2
\end{aligned}
\tag{4.5}
$$

The equality (i) uses the fact that $\boldsymbol{\varepsilon}$ has zero mean. The equality (ii) expands out $\boldsymbol{V}$ as column vectors $[\boldsymbol{v}_1, \boldsymbol{v}_2]$ and expresses $\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\intercal] = \pi\pi^\intercal = \begin{bmatrix} p \\ 1-p \end{bmatrix}\begin{bmatrix} p & 1-p \end{bmatrix}$. Rearranging the expressions yields equality (iii).

Therefore, we can equivalently penalize the Frobenius norm of Γ and l_2 norm of the vector $\boldsymbol{V}\pi$, i.e. $||\Gamma||_F^2 + ||\boldsymbol{V}\pi||_2^2$, instead of taking the empirical mean of the demand difference when performing the regularization. To summarize, the data generator determines the filter weight $\boldsymbol{G}$ and outputs the perturbed demand $\tilde{\boldsymbol{d}}$, while the adversary takes in the altered demand $\tilde{\boldsymbol{d}}$ and private labels y to try to learn a classifier.

4.3.3 Minimax learning

We construct two neural networks to perform the roles of the two players, one is for the data generator and the other one is for the adversary. To train the adversary, we minimize the cross-entropy loss $\mathcal{L}_a$, i.e. $\min_\psi \mathcal{L}_a\big(f_\psi(\tilde{\boldsymbol{d}}), \mathrm{y}\big)$, which follows the loss function mentioned in section 4.3.1. For the generator, we decouple the training into two steps. First, we leverage the loss that is passed from the adversary to update the

matrix weight $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$, i.e.

$$(\textbf{step1}) \quad \min_{\boldsymbol{G}} -\lambda_a \mathcal{L}_a\Big(f_\psi\big(\boldsymbol{d} + \Gamma\boldsymbol{\varepsilon} + \boldsymbol{V}\boldsymbol{y}\big), \boldsymbol{y}\Big) + \kappa\big(||\Gamma||_F^2 + ||\boldsymbol{V}\pi||_2^2\big), \tag{4.6}$$

where κ is the hyper-parameter that penalizes the distance between $\tilde{d}$ and d implicitly. The next step is to use the privatized demand $\tilde{\boldsymbol{d}} = \boldsymbol{d} + \widehat{\boldsymbol{G}} \begin{bmatrix} \varepsilon \\ \boldsymbol{y} \end{bmatrix}$ to determine the control by running the following optimization:

$$\begin{aligned}(\textbf{step2}) \quad \arg\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(0,\boldsymbol{I})}\Big\{&\boldsymbol{p}^\intercal\big(\boldsymbol{x}_{in} - \boldsymbol{x}_{out} + \tilde{\boldsymbol{d}}\big)_+ \\ &+ \beta_1||\boldsymbol{x}_{in}||_2^2 + \beta_2||\boldsymbol{x}_{out}||_2^2 + \beta_3||\boldsymbol{x}_s - \alpha B||_2^2\Big\}\end{aligned} \tag{4.7a}$$

The optimal solution of the above convex problem (4.7) is $\tilde{\boldsymbol{x}}^*$, or more specifically $\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}})$, because it is a function of the privatized demand, which is aligned with equation (4.3c). The third step calculates the loss, $\mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d})$, using $\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}})$ and the original raw demand expressed as:

$$\begin{aligned}(\textbf{step3}) \quad \mathcal{L}_u(\tilde{\boldsymbol{x}}^*(\tilde{\boldsymbol{d}}), \boldsymbol{d}) = &\boldsymbol{p}^\intercal\Big(\tilde{\boldsymbol{x}}^*_{in}(\tilde{\boldsymbol{d}}) - \tilde{\boldsymbol{x}}^*_{out}(\tilde{\boldsymbol{d}}) + \boldsymbol{d}\Big)_+ \\ &+ \beta_1||\tilde{\boldsymbol{x}}^*_{in}||_2^2 + \beta_2||\tilde{\boldsymbol{x}}^*_{out}||_2^2 + \beta_3||\tilde{\boldsymbol{x}}^*_s - \alpha B||_2^2.\end{aligned} \tag{4.8a}$$

We update $\boldsymbol{G}$ using gradient descent with the gradient determined by the chain rule. Recall that the generator outputs a privatized demand with reduced correlation to the sensitive label that is also used to yield the storage control decisions. Those decisions are evaluated on the cost given the raw demand, thus, the Jacobian of $\boldsymbol{G}$ is

$$g_{\boldsymbol{G}} = \nabla_{\boldsymbol{G}}\mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d}) = \frac{\partial \mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \tilde{\boldsymbol{d}}} \frac{\partial \tilde{\boldsymbol{d}}}{\partial \boldsymbol{G}}. \tag{4.9}$$

In the context of our storage control problem, the first term in (4.9) is

$$\frac{\partial \mathcal{L}_u(\boldsymbol{x}, \boldsymbol{d})}{\partial \boldsymbol{x}} = \begin{cases} \boldsymbol{Q}\boldsymbol{x} + \begin{bmatrix} \boldsymbol{p} \\ -\boldsymbol{p} \\ \boldsymbol{0} \end{bmatrix}, & \text{if } \boldsymbol{D}\boldsymbol{x} - \boldsymbol{d} > 0 \\ \boldsymbol{Q}\boldsymbol{x} & \text{otherwise} \end{cases}, \tag{4.10}$$

where $\boldsymbol{Q}$ is given in the Appendix equation (4.21), $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$.

The second term, i.e. $\frac{\partial \boldsymbol{x}}{\partial \tilde{\boldsymbol{d}}}$, in (4.9) hinges on automatic differentiation through a convex program[157, 158]. Because an optimization problem can be viewed as a function mapping the problem data to the primal and dual solutions, we can convert problem (4.7) to a conic form and calculate the changes of the optimal solution given the perturbations of the problem data. The transformed formulation leverages the idea of finding a zero solution for the residual map of a homogeneous self-dual embedding derived from the KKT conditions of the convex program[158, 159, 160].

The third term in (4.9) is

$$\mathsf{d}\boldsymbol{G} := \frac{\partial \tilde{\boldsymbol{d}}}{\partial \boldsymbol{G}} = \begin{bmatrix} \frac{\mathsf{d}\tilde{\boldsymbol{d}}}{\varepsilon_1} & \dots & \frac{\mathsf{d}\tilde{\boldsymbol{d}}}{\varepsilon_H} & \frac{\mathsf{d}\tilde{\boldsymbol{d}}}{p} & \frac{\mathsf{d}\tilde{\boldsymbol{d}}}{1-p} \end{bmatrix} \in \mathbb{R}^{H \times (H+2)}, \tag{4.11}$$

since $\mathsf{d}\tilde{\boldsymbol{d}} = \mathsf{d}\boldsymbol{G} \begin{bmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{y} \end{bmatrix}$. Thus, all three terms in equation (4.9) can be evaluated in the backward pass of the generator training and we can update the filter weight $\boldsymbol{G}$ using stochastic gradient decent[161]: $\boldsymbol{G}_{k+1} := \boldsymbol{G}_k - \eta_l g_{\boldsymbol{G}}$ where k is the iteration step and η_l is the learning rate.

Remark: To summarize, Step 1 shown in equation (4.6) updates the matrix $\boldsymbol{G}$ by minimizing the negative classification loss (equivalent to maximizing the classification loss) of the adversary, while maintaining the constraint determined in (4.5). Step 2 calculates the optimal control of the storage using the privatized demand. In Step 3, $\boldsymbol{G}$ is updated by evaluating the gradient of the energy cost given the control based

on the privatized demand. The updates are expressed as

$$(\text{update1})\widehat{\boldsymbol{G}}_{k+1} = \boldsymbol{G}_k - \eta_l^{(k)} \nabla_G \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y}) \tag{4.12a}$$

$$(\text{update2})\boldsymbol{G}_{k+1} = \widehat{\boldsymbol{G}}_{k+1} - \eta_l^{(k)} \nabla_G \mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d}) \tag{4.12b}$$

$$(\text{adversary update})\psi_{k+1} = \psi_k - \eta_l \nabla_\psi \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y}), \tag{4.12c}$$

which run until convergence. We set the learning rates in each step to be equal for simplicity. The training procedure is described in Algorithm 1.

Algorithm 1: Minimax learning

Input: Demand data $\mathcal{D}$, label data $\mathcal{Y}$, learning rate η_l, parameters $\{B, \alpha, \beta_1, \beta_2, \beta_3\}$, and hyper-parameters κ_1, κ_2

Initialize $\boldsymbol{G}_k$, ψ_k at iteration $k = 0$ with batch size m;

while *ψ or $\boldsymbol{G}$ has not converged* **do**

  draw batches of pair $(\boldsymbol{d}^{(i)}, \boldsymbol{y}^{(i)})$ from demand and label datasets $(\mathcal{D}, \mathcal{Y})$, $\forall i = 1, \ldots, m$;

  Sample batch of Gaussian random vectors $\boldsymbol{\varepsilon}^{(1),\ldots,(m)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$;

  $\psi_{k+1} := \psi_k - \eta_l \mathbb{E}[\nabla_\psi \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y})]$;

  $\widehat{\boldsymbol{G}}_{k+1} := \boldsymbol{G}_k - \eta_l \mathbb{E}[\nabla_G \mathcal{L}_a(f_\psi(\tilde{\boldsymbol{d}}), \boldsymbol{y})]$;

  $\boldsymbol{G}_{k+1} := \widehat{\boldsymbol{G}}_{k+1} - \eta_l \mathbb{E}[\nabla_G \mathcal{L}_u(\tilde{\boldsymbol{x}}^*, \boldsymbol{d})]$ where $\tilde{\boldsymbol{x}}^*$ is optimal solution of (4.7)

  (The expected gradient value is approximated as the sample mean of the batch.)

end

return $\boldsymbol{G}$ and ψ

4.3.4 Convergence of the filter

This subsection focuses on the stability and boundedness of the iterates in our back-propagation that leverage stochastic gradient methods (or some related variants of first-order gradient methods). Using the subgradient property [122, Chapter 9.1], g is a subgradient of f at x if

$$f(y) \geq f(x) + \langle g, y - x \rangle \quad \forall y, \tag{4.13}$$

and assuming $\boldsymbol{G}^*$ is a local optimal point; when we apply the step1 and step3 updates $\boldsymbol{G}_{k+1} = \boldsymbol{G}_k - \eta_l^{(k)} \nabla \mathcal{L}_a^{(k)} - \eta_l^{(k)} \mathcal{L}_u^{(k)}$ at the k-th iteration, we can obtain the following relationship

$$\mathbb{E}[||\boldsymbol{G}_{k+1} - \boldsymbol{G}^*||_2^2] \tag{4.14a}$$

$$= \mathbb{E}[||\boldsymbol{G}_k - \eta_l^{(k)}(\nabla \mathcal{L}_a^{(k)} + \nabla \mathcal{L}_u^{(k)}) - \boldsymbol{G}^*||_2^2] \tag{4.14b}$$

$$= \mathbb{E}[||\boldsymbol{G}_k - \boldsymbol{G}^*||_2^2] - 2\eta_l^{(k)} \mathbb{E}\langle \nabla \mathcal{L}_a^{(k)} + \nabla \mathcal{L}_u^{(k)}, \boldsymbol{G}_k - \boldsymbol{G}^* \rangle + (\eta_l^{(k)})^2 \underbrace{||\nabla \mathcal{L}_a^{(k)} + \nabla \mathcal{L}_u^{(k)}||_2^2}_{\delta_k^2} \tag{4.14c}$$

$$\overset{\text{(i)}}{=} \mathbb{E}[||\boldsymbol{G}_k - \boldsymbol{G}^*||_2^2] - 2\eta_l^{(k)} \mathbb{E}\langle \nabla \mathcal{L}_a^{(k)}, \boldsymbol{G}_k - \boldsymbol{G}^* \rangle - 2\eta_l^{(k)} \mathbb{E}\langle \nabla \mathcal{L}_u^{(k)}, \boldsymbol{G}_k - \boldsymbol{G}^* \rangle + (\eta_l^{(k)})^2 \delta_k^2 \tag{4.14d}$$

$$\overset{\text{(ii)}}{\leq} \mathbb{E}[||\boldsymbol{G}_k - \boldsymbol{G}^*||_2^2] - 2\eta_l^{(k)} \big(\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*\big) - 2\eta_l^{(k)} \big(\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*\big) + (\eta_l^{(k)})^2 \delta_k^2. \tag{4.14e}$$

Equality (i) expands the inner product of the loss gradients and iterates using δ_k for the norm of the sum of loss gradients. The inequality (ii) uses the subgradient condition in equation (4.13), $\mathcal{L}(\boldsymbol{G}_k) - \mathcal{L}(\boldsymbol{G}^*) \geq \langle \nabla \mathcal{L}^{(k)}, \boldsymbol{G}_k - \boldsymbol{G}^* \rangle$ (both for $\mathcal{L}_a$ and $\mathcal{L}_u$). Rearranging equation (4.14a) and equation (4.14e), we get

$$\begin{aligned} &2\eta_l^{(k)} \big(\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*\big) + 2\eta_l^{(k)} \big(\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*\big) \\ &\leq \mathbb{E}[||\boldsymbol{G}_k - \boldsymbol{G}^*||_2^2] - \mathbb{E}[||\boldsymbol{G}_{k+1} - \boldsymbol{G}^*||_2^2] + (\eta_l^{(k)})^2 \delta_k^2. \end{aligned} \tag{4.15}$$

By summing iterates up to step K, we get

$$2\Big(\sum_{k=1}^{K} \eta_l^{(k)}\Big) \min_{k \in [k]} [\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*] + \min_{k \in [k]} [\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*] \tag{4.16a}$$

$$\overset{\text{(iii)}}{\leq} 2 \sum_{k=1}^{K} \eta_l^{(k)} [\mathcal{L}_a(\boldsymbol{G}_k) - \mathcal{L}_a^*] + [\mathcal{L}_u(\boldsymbol{G}_k) - \mathcal{L}_u^*] \tag{4.16b}$$

$$\overset{\text{(iv)}}{\leq} ||\boldsymbol{G}_1 - \boldsymbol{G}^*||_2^2 + \sum_{k=1}^{K} (\eta_l^{(k)})^2 \delta_k^2 \tag{4.16c}$$

where the inequality (iii) is valid since we take the minimum over all iterations and inequality (iv) is derived from the summation of equation (4.15). Then, arranging equation (4.16a) and equation (4.16c) gives

$$\begin{aligned} &\min_{k\in[k]}[\mathcal{L}_1(\boldsymbol{G}_k) - \mathcal{L}_1^*] + \min_{k\in[k]}[\mathcal{L}_2(\boldsymbol{G}_k) - \mathcal{L}_2^*] \\ &\leq \frac{||\boldsymbol{G}_1 - \boldsymbol{G}^*||_2^2 + \sum_{k=1}^K (\eta_l^{(k)})^2 \delta_k^2}{2\sum_{k=1}^K \eta_l^{(k)}} \end{aligned} \tag{4.17a}$$

Thus, if the 2-norm of the vectorized version of $\boldsymbol{G}_1 - \boldsymbol{G}^*$ is bounded by r, and with learning rate $\sum_k \eta_l^{(k)} \to \infty$ but $\sum_k (\eta_l^{(k)})^2 < \infty$, the right hand-side of equation (4.17a) becomes $\frac{r^2 + \sum_k (\eta_l^{(k)})^2 \delta_k^2}{2\sum_k \eta_l^{(k)}} \to 0$. Therefore, using the gradient updates in step1 and step3 minimizes the losses $\mathcal{L}_a, \mathcal{L}_u$ and converges to a local optimal point.

4.4 Experiments

In this section, we evaluate the capability of our linear filter to (1) generate perturbed smart meter data that reduces the prediction accuracy of sensitive attributes; (2) maintain the minimum energy cost from an optimal control decision using the perturbed data; (3) integrate into a contemporary deep learning architecture with parallelism. The code for our experiments is available at https://github.com/markcx/DER_ControlPrivateTimeSeries.

4.4.1 Setup

We build up two neural networks to form the adversarial classifier and generator. The adversarial classifier is composed of two fully connected layers with ELU (Exponential Linear Unit) activation to estimate the sensitive attribute from demand. The first layer contains the same number of neurons as the time steps of the meter data series used by the battery optimal controller, and the second layer has half of the neuron numbers of the first layer and outputs a two dimensional vector representing the probability of the associated categories of the label. The generator module is composed of a single linear layer that takes a standard normal random vector and

the private labels as inputs, and outputs noise to be added to the original demand. The parameters of the single linear layer form matrix $\boldsymbol{G}$. Additionally, we specify $\boldsymbol{G}$ to be block diagonal to reduce the number of learning parameters, i.e. $\boldsymbol{G} = [\Gamma, \boldsymbol{V}]$ where Γ is a diagonal matrix. Given the number of columns in our weight matrix is c_w (e.g. the c_w for $\boldsymbol{G}$ is 26 for the solar dataset and 50 in our residential experiments), we use uniform initialization[162] between $(-\frac{1}{c_w}, \frac{1}{c_w})$ for both the adversary and generator networks. We use 85% of the data for training and the remaining 15% for testing the performance of the filter. Later in section 4.4.4, we demonstrate that our method is robust to different training and testing splits. We set hyper-parameters $\beta_1 = \beta_2 = \beta_3 = 10^{-5}, \kappa = 10^{-3}$ throughout the experiments. The learning rate for the classifier is 10^{-3} and the learning rate for the generator starts from 0.1 and decays 20% for every 100 steps. We present the classification accuracy to indicate the correlation, as a lower accuracy implies a lower value of mutual information[15], thus, there is less correlation between the demand and sensitive labels. We set the initial battery state of charge to 1% of its maximum energy capacity, i.e. $B_{init} = 0.01B$. We use a time-of-use price structure with two tiers: a high price of \$0.463 per KWh from 4pm-9pm and \$0.202 per KWh for the rest of the day.

4.4.2 Examples

Deployment of storage on aggregated demand with solar generation

For our first experiment, we aggregated 24-hour demand consumption from thousands of homes into groups of 100-200 homes and added solar generation. The aggregations represent the demand seen at a secondary transformer from the perspective of a utility company. The goal is to minimize the energy cost for the aggregation of homes by running the optimal charging and discharging controls for battery storage located at the secondary transformer given a prescribed price. Before the experiment, each demand profile is assigned a binary label indicating if it is from a high- or low-income group, with high-income groups having a peak demand above a certain threshold. During the experiment, we wish to privatize the demand before sending it to the storage operator to perform cost minimization, so the operator cannot infer whether

the aggregation of customers comes from a high or low-income group. The upper panel of Figure 4.1 shows the income attribute can be easily inferred from the raw demand as the height of the peaks are clearly distinguishable. The lower panel of Figure 4.1 shows that the privatized demands are perturbed such that two labels overlap making it harder to tell which demand has high or low income. However, there is a trade-off between privacy and utility when perturbing the data. We use the hyper-parameter λ_a to balance the adversarial loss and the utility loss i.e. smaller λ_a means less weight for privacy and more for utility, as shown in Figure 4.2. When λ_a increases from 8 to 128, the classification accuracy of the income label drops from 89.4% to 73% as we expected. The raw classification accuracy with zero weight is 95.2%. The loss of performance of the cost minimization by using privatized demand instead of raw demand ranges from 2.5% at $\lambda_a = 8$ to almost 5% at $\lambda_a = 128$ on average, which shows that high privacy comes with a performance cost for this battery control problem.

Deployment of storage on residential users

The second experiment considers residential customers adopting batteries to minimize their energy cost without selling excess to the grid. The control of the battery is performed by an outside program, so the owner wishes to privatize their demand before sending it to the controller. The dataset is from the Irish CER Smart Metering Project[163, 6]. We select a year of meter data for meters that contain a record indicating if they belong to a large or small home and partition it into daily sequences with 48 entries for each day. We end up with 54478 records in total. Recall that our goal is to create altered demand that won't degrade the cost savings while removing the correlation between the demand and the attribute indicating a small or large home. Differences between this experiment and the previous one are that this experiment uses data from only a single home versus an aggregation of homes, and this experiment uses real world labeled data instead of synthetic labels. Figure 4.3 depicts the trade-off between utility degradation and privacy gain for different weights on privacy loss. The accuracy of classifying large or small homes based on the raw demand is 77.5%. When we have low weight on the privacy loss (e.g. $\lambda_a = 0.5$),

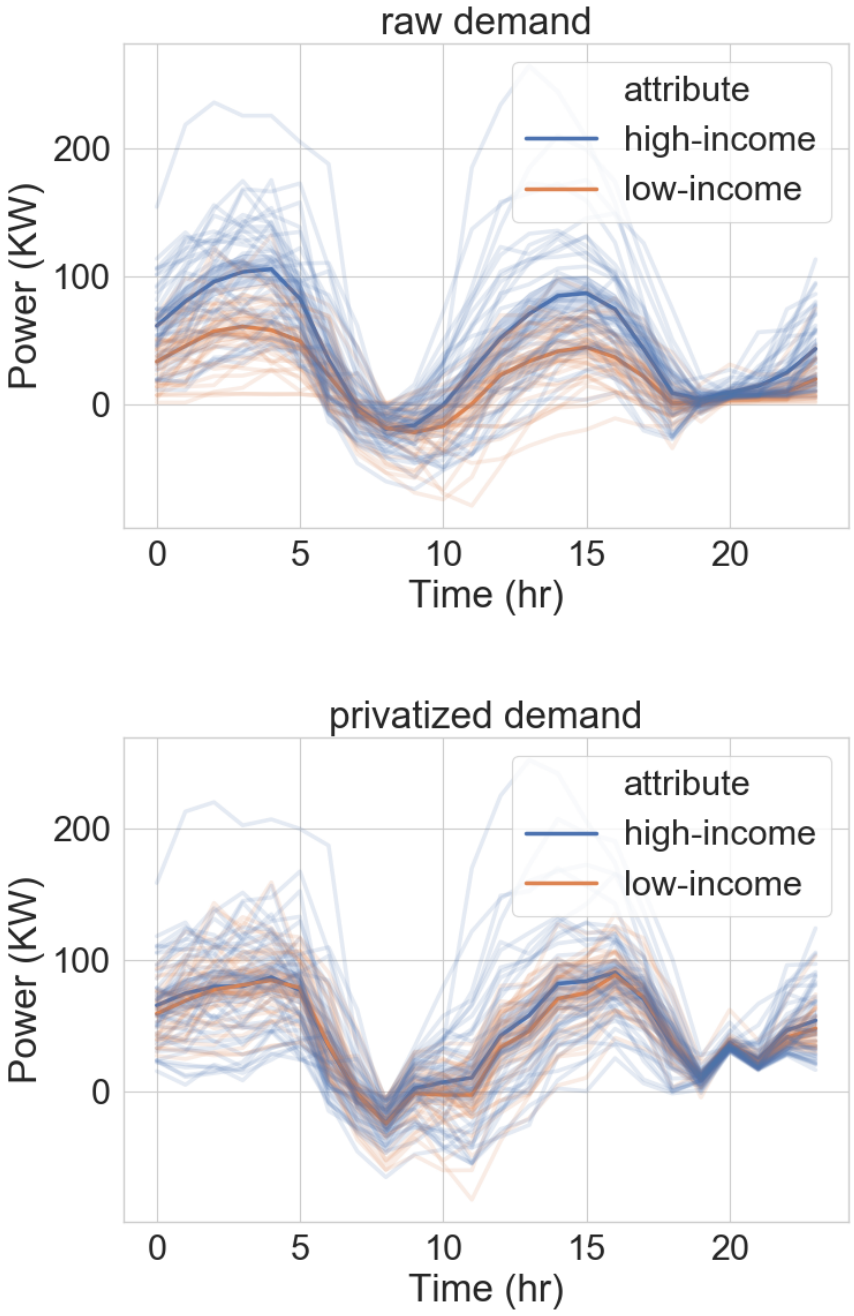


Figure 4.1: A batch of 24-hour demand with solar generation that is net negative in certain hours allowing storage to minimize the cost through an optimal charge and discharge sequence. The **upper panel** shows the raw demand. The **lower panel** shows the privatized demand.

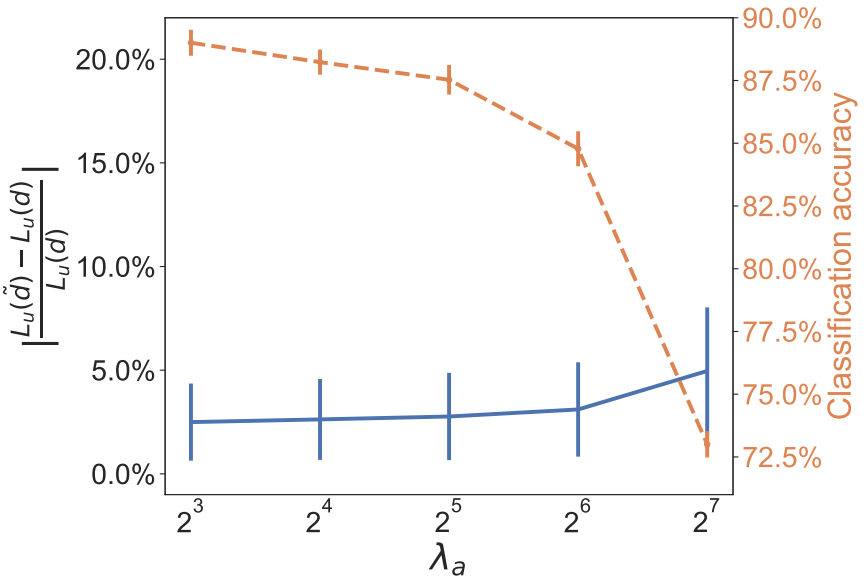


Figure 4.2: The trade-off between privacy and utility controlled by parameter λ_a, which places weight on the private attribute classification loss.

the classification accuracy only drops a little to 75%, with a greater sacrifice on cost saving performance (e.g. increased to 8% more cost on average). In the high privacy weight scenario, the classification accuracy drops down to 50% as desired, while the utility performance gap only increases up to 12%. When comparing this experiment to the previous one, we find that the adversary has more difficulty determining home size for individual homes than for aggregations of homes with comparable loss of cost minimization performance.

Integration into real world systems

This approach can be integrated into existing storage control systems such as those proposed in [134, 135] in the following manner. First, the privacy filter is trained offline using an anonymous batch of private data from many sources before the installation of the storage system and control algorithm. Then, the learned filter weights are given to the data owner who wishes to use the system. Next, during operation, the data owner locally privatizes the power demand data by locally computing the

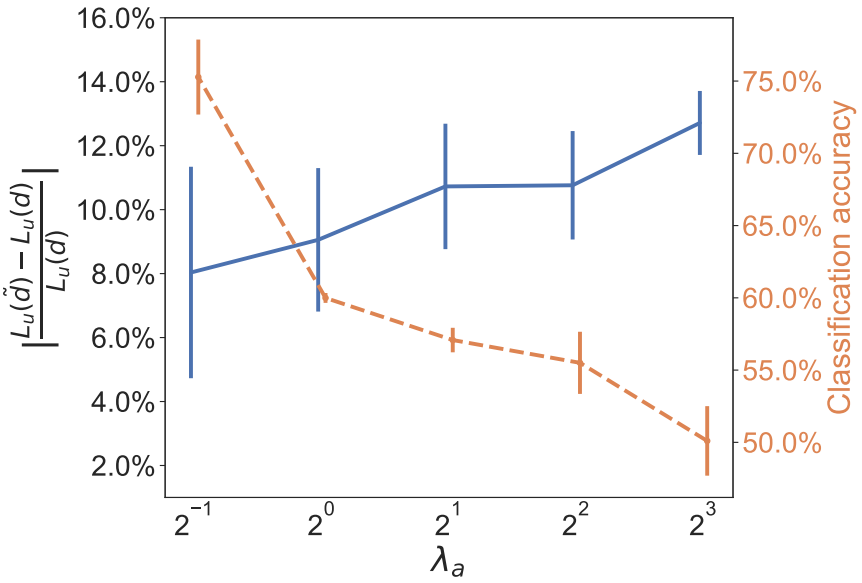


Figure 4.3: The trade-off between the utility and privacy for the CER dataset[163]. The privacy label indicates a large or small home. λ_a weighs the privacy loss.

matrix product between the learned filter weights and the power demand data. The matrix product can be computed locally with minimal computation since the filter weight matrix is diagonal. Finally, the storage control algorithm receives the privatized data that is computed locally and performs the cost minimization optimization on the privatized data just as it would with raw data.

4.4.3 Parallelism

The training for the experiments in this section are run on a six-core Intel Core i7 CPU @2.2GHz. Current standard solvers like Gurobi or Mosek without support of in-batch parallelism can be computationally expensive for solving a quadratic problem. Our filter makes use of automatic differentiation for a cone program (DIFFCP)[158] and leverages multiprocessing to speed up the forward and backward calculations.

Figure 4.4 displays the mean and standard deviation of running each trial 8 times, showing that our batched module outperforms Gurobi or Mosek, which are highly tuned commercial solvers for reasonable batch sizes. For a minibatch size of 128, we

solve all problems in an average of 1.31 seconds, whereas Gurobi takes an average of 11.7 seconds. This speed improvement for a single minibatch makes the difference between a practical and an unusable solver in the context of training a deep learning architecture.

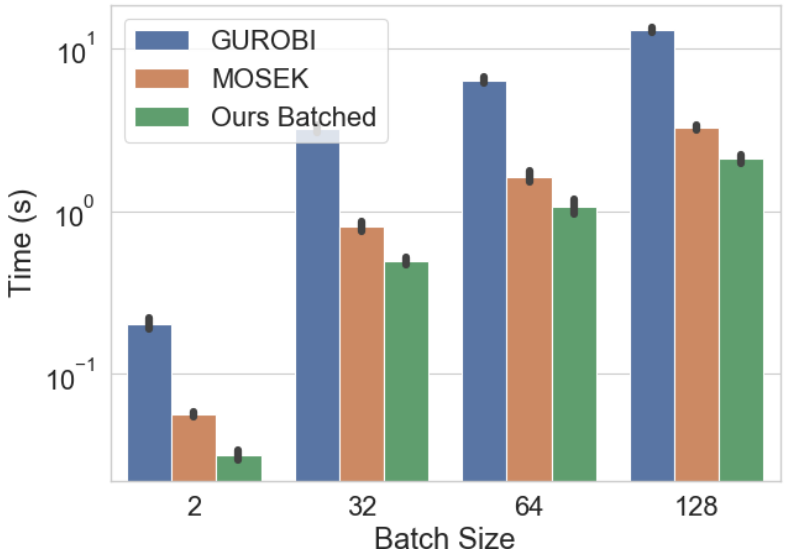


Figure 4.4: CPU run time of a batched optimization using Gurobi v8.1.0, Mosek v8.1.0.60, and our parallel module.

4.4.4 Sensitivity analysis

In this section, we evaluate the sensitivity of our method to: (i) the inherent trade-off between data privacy and utility, and (ii) the ratio of training data to testing data. First, we summarize our findings on the trade-off between data privacy and utility. As discussed, the tunable hyper-parameter, λ_a, allows us to scale the importance of privacy. In the first example, when λ_a increases from 8 to 128, the classification accuracy of the income label drops from 89.4% to 73% while loss of performance of the cost minimization increases from 2.5% to almost 5% on average as seen in Figure 4.2. In the second example, when λ_a increases from 0.5 to 4, the classification accuracy drops from 75% to 50% while loss of cost saving performance increases from 8% to

12% as seen in Figure 4.3. These performance values represent a Pareto optimal set parameterized by λ_a with the best point depending on the specific external values assigned to privacy and utility for the given scenario.

Here, we demonstrate the robustness of our method to the training data by evaluating the performance of battery control on the residential dataset via various ratios of training/testing split with fixed $\lambda_a = 2$. The results are shown in Table 4.1. We find that the classification accuracy of the private attribute and the sacrificed cost gap are consistently around 57-61% and 9-12% respectively. This difference is small compared to other sources of variation such as the choice of λ_a, and comparable to the variation seen from different batches within the data. Such a result indicates our approach is relatively robust to different training and testing splits of the dataset.

Table 4.1: Evaluation of performance on various train/test splits of the Irish CER data when $\lambda_a = 2$.

train/test(%)	baseline	65/35	70/30	75/25	80/20	85/15
acc. (%)†	77.5	63.3	61.1	57.6	58.5	56.9
cost gap (%)*	0	11.7	9.4	12.2	10.0	10.9

† Accuracy of private attribute. * Gap above optimal energy cost of controlling batteries with raw data. Lower values are preferred for accuracy and optimal objective gap.

4.5 Conclusion

We have presented a method for the privatization of personal data that maintains its utility in the optimal control of energy resources. Our method comprises a small linear filter that adds random noise to the data conditional on the private attributes we wish to protect. The linear filter is trained using a minimax optimization procedure that balances the trade-off between classifcation accuracy of the private attributes and the performance of an optimal controller. Additionally, we include a distortion penalty to preserve aspects of the data that are not specified by the utility or privacy functions in order to avoid adding arbitrary noise. We have demonstrated that this method is effective in two datasets and easy to integrate into real world DER control

solutions. In the first dataset on aggregations of homes, the private label accuracy dropped by 26% while the utility performance gap only increased by 5%. The second dataset on individual homes saw the classification accuracy for the binary label drop down to the minimum of 50%, while the utility performance gap only increased up to 12%. Limitations of this method include the requirement to solve an optimization in the training loop, which can be computationally intensive for large problems; however, we suspect only a few iterations of the optimization are needed to achieve the desired gradients, which will dramatically reduce the computation required. Future work will look intro reducing the training computation time with fewer optimization iterations, increasing the variety of experiments with additional private labels and utility optimizations, and the consideration of additional noise due to poor data quality.

4.6 Supplements

4.6.1 Battery control details

We present a snapshot of the results for the storage control based on raw and private demand data. Figure 4.5 displays the storage control for our experiment with aggregated homes and solar generation. The upper-left and lower-left panel show the 24-hour charging and discharging decisions with each color representing one sample in a batch. The control decisions made with raw versus privatized demand data are closely aligned in general, but have different charging and discharging amounts of power due to perturbation. However, such an altered charging profile doesn't increase the minimum cost of energy too much as we can see from the upper-right and lower-right panels of Figure 4.5. The electricity cost increases by a maximum of $22 USD per day given that the highest daily cost is around US $390 USD. (Each bin spans the range of $2.5 USD for Figure 4.5.) Figure 4.6 shows the same information, but for the second experiment on individual home data.

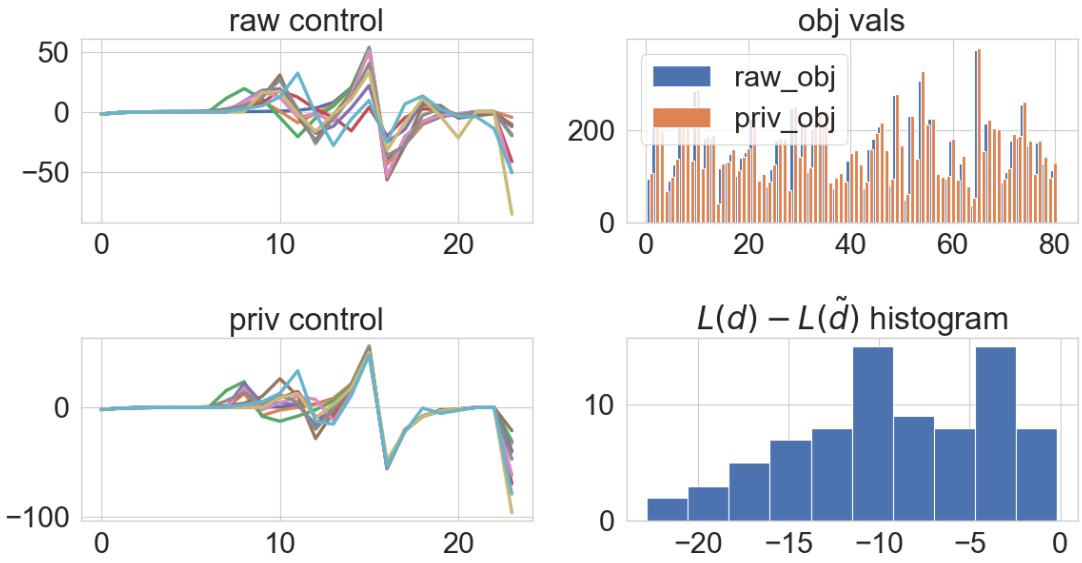


Figure 4.5: Analysis of storage control for the aggregated homes experiment with $\lambda_a = 128$. The **upper-** and **lower-left** panel show the charging and discharging power in kilowatts (KW). Different colored curves represent different samples in the batch. The **upper-right** panel shows the daily electricity cost when operating the battery using raw or private demand (x-axis is the sample number, y-axis is in dollars ($)). The **lower-right** panel shows a histogram of the loss gap. (The x-axis is the increased cost in $; the y-axis is the number of days that show similar cost increases in a batch.)

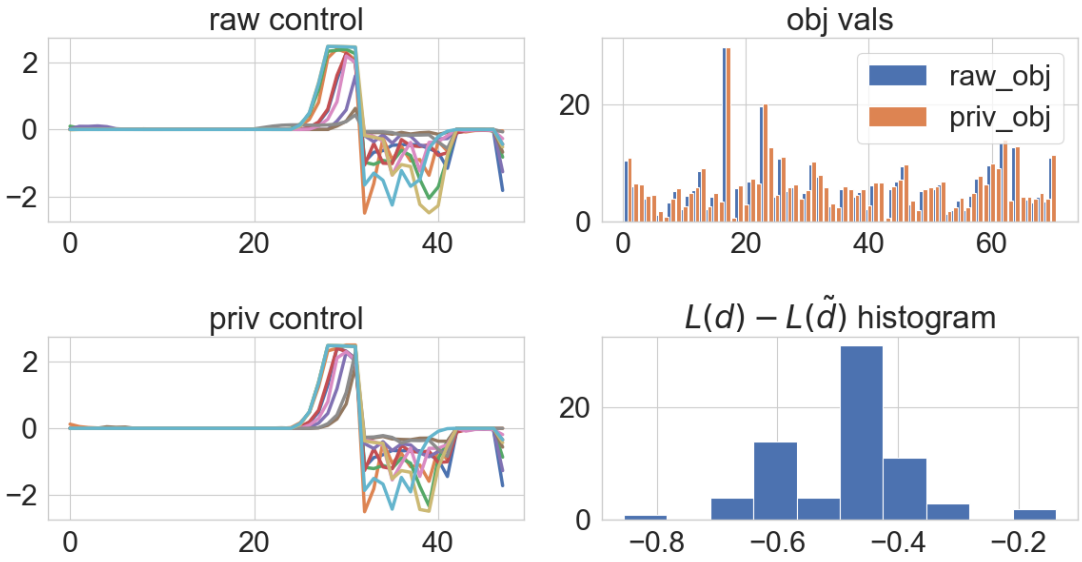


Figure 4.6: Analysis of storage control for the CER data experiment with $\lambda_a = 8$. Each panel has the same x- and y-axis as Figure 4.5

4.6.2 Quadratic optimization

A canonical form of the quadratic constrained minimization problem (QP) is expressed as follows:

$$\min_x \quad \frac{1}{2}x^T Q x + q^T x \tag{4.18a}$$

$$\text{s.t} \quad Ax = b \tag{4.18b}$$

$$Gx \leq h. \tag{4.18c}$$

We first show that the basic battery storage problem can be considered as a special case of QP. We start with the 24-hour horizon storage problem in Problem4.1. We

can express the constraints from equation (4.1d) to equation (4.1f) as

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ -I & 0 & 0 \\ 0 & I & 0 \\ 0 & -I & 0 \\ 0 & 0 & I \\ 0 & 0 & -I \\ -I & I & 0 \end{bmatrix}}_{G} \begin{bmatrix} x_{in} \\ x_{out} \\ x_s \end{bmatrix} \leq \begin{bmatrix} c_{in} \\ 0 \\ c_{out} \\ 0 \\ B \\ 0 \\ \boldsymbol{d} \end{bmatrix} \Leftrightarrow Gx \leq h. \tag{4.19}$$

We add a constraint that the net of the demand and storage is greater than or equal to 0, so we can formulate the objective as a QP. This constraint does not modify the original problem as long as it is feasible because the optimal solution will implicitly make the net of demand and storage greater than or equal to 0. The constraints in equation (4.1b)-equation (4.1c) are expressed as

$$\underbrace{\begin{bmatrix} 0 & 0 & 1, \dots 0 \\ [\eta_{in} I, 0] & [-1/\eta_{out} I, 0] & [I, 0] - [0, I] \end{bmatrix}}_{A} \begin{bmatrix} x_{in} \\ x_{out} \\ x_s \end{bmatrix} = \begin{bmatrix} B_{init} \\ 0 \end{bmatrix} \tag{4.20}$$

$$\Leftrightarrow Ax = b,$$

with $[I, 0] \in \mathbb{R}^{23 \times 24}$. The objective equation (4.1a) can be converted to a standard QP by letting

$$\boldsymbol{Q} = \begin{bmatrix} \beta_1 I & 0 & 0 \\ 0 & \beta_2 I & 0 \\ 0 & 0 & \beta_3 I \end{bmatrix}, \quad q = \begin{bmatrix} p \\ -p \\ -2\beta_3 \alpha B \mathbf{1} \end{bmatrix}. \tag{4.21}$$

Therefore, it is straightforward to discover that $x^T Q x + q^T x$ is the new form of the objective.

Chapter 5

Fair Selection of Customers in Demand Response

Chapter 6

Conclusions and Discussions

...

Appendix A

Deferred Content

...

Bibliography

[1] Brendan J Kirby. *Spinning reserve from responsive loads*. Citeseer, 2003.

[2] Duncan S Callaway and Ian A Hiskens. Achieving controllability of electric loads. *Proceedings of the IEEE*, 99(1):184–199, 2011.

[3] Victoria Y Pillitteri and Tanya L Brewer. Guidelines for smart grid cybersecurity. *National Institute of Standards and Technology*, 2014.

[4] Matthew P Barrett. Framework for improving critical infrastructure cybersecurity. *National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep*, 2018.

[5] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[6] Christian Beckel, Leyna Sadamori, Thorsten Staake, and Silvia Santini. Revealing household characteristics from smart meter data. *Energy*, 78:397–410, 2014.

[7] Douglas S Massey. The legacy of the 1968 fair housing act. In *Sociological Forum*, volume 30, pages 571–588. Wiley Online Library, 2015.

[8] John R Nevin and Gilbert A Churchill Jr. The equal credit opportunity act: An evaluation. *Journal of Marketing*, 43(2):95–104, 1979.

[9] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

[10] Lee V White and Nicole D Sintov. Health and financial impacts of demand-side response measures differ across sociodemographic groups. *Nature Energy*, 5(1):50–60, 2020.

[11] Jamal Lewis, Diana Hernández, and Arline T Geronimus. Energy efficiency as energy justice: addressing racial inequities through investments in people and places. *Energy efficiency*, 13(3):419–432, 2020.

[12] Constructing dynamic residential energy lifestyles using latent dirichlet allocation. https://web.stanford.edu/~markcx/sample-project/LDA_Energy_Lifestyle_manuscript.pdf. Accessed: 2021-11-03.

[13] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Context-aware generative adversarial privacy. *Entropy*, 19(12):656, 2017.

[14] Xiao Chen, Peter Kairouz, and Ram Rajagopal. Understanding compressive adversarial privacy. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6824–6831. IEEE, 2018.

[15] Xiao Chen, Thomas Navidi, Stefano Ermon, and Ram Rajagopal. Distributed generation of privacy preserving data with user customization. *Safe Machine Learning workshop at ICLR*, 2019.

[16] Xiao Chen, Thomas Navidi, and Ram Rajagopal. Energy resource control via privacy preserving data. *Electric Power Systems Research*, 189:106719, 2020.

[17] A. Ghosal and M. Conti. Key management systems for smart grid advanced metering infrastructure: A survey. *IEEE Communications Surveys Tutorials*, 21(3):2831–2848, 2019.

[18] F. Fahiman, S. M. Erfani, S. Rajasegarar, M. Palaniswami, and C. Leckie. Improving load forecasting based on deep learning and k-shape clustering. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4134–4141, 2017.

[19] Edward Barbour and Marta González. Enhancing household-level load forecasts using daily load profile clustering. In *Proceedings of the 5th Conference on Systems for Built Environments*, BuildSys '18, page 107–115, New York, NY, USA, 2018. Association for Computing Machinery.

[20] B. Yildiz, J. I. Bilbao, J. Dore, and A. Sproul. Household electricity load forecasting using historical smart meter data with clustering and classification techniques. In *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pages 873–879, 2018.

[21] J. Kwac and R. Rajagopal. Data-driven targeting of customers for demand response. *IEEE Transactions on Smart Grid*, 7(5):2199–2207, 2016.

[22] Jeffrey Wong and Ram Rajagopal. A simple way to use interval data to segment residential customers for energy efficiency and demand response program targeting. In *ACEEE Proceedings*, pages 1–9, 2012.

[23] Annika Todd-Blick, C Anna Spurlock, Ling Jin, Peter Cappers, Sam Borgeson, Dan Fredman, and Jarett Zuboy. Winners are not keepers: Characterizing household engagement, gains, and energy patterns in demand response using machine learning in the united states. *Energy Research & Social Science*, 70:101595, 2020.

[24] Amir Kavousian, Ram Rajagopal, and Martin Fischer. Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants' behavior. *Energy*, 55:184–194, 2013.

[25] Valeria Di Cosmo and Denis O'Hora. Nudging electricity consumption using tou pricing and feedback: evidence from irish households. *Journal of Economic Psychology*, 61:1–14, 2017.

[26] Milad Afzalan and Farrokh Jazizadeh. Residential loads flexibility potential for demand response using energy consumption patterns and user segments. *Applied Energy*, 254:113693, 2019.

[27] Ivana Dusparic, Adam Taylor, Andrei Marinescu, Fatemeh Golpayegani, and Siobhan Clarke. Residential demand response: Experimental evaluation and comparison of self-organizing techniques. *Renewable and Sustainable Energy Reviews*, 80:1528–1536, 2017.

[28] Hilary Boudet, Chad Zanocco, Greg Stelmach, Mahmood Muttaqee, and June Flora. Public preferences for five electricity grid decarbonization policies in california. *Review of Policy Research*, 2021.

[29] Lucien Werner, Adam Wierman, and Steven H Low. Pricing flexibility of shiftable demand in electricity markets. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pages 1–14, 2021.

[30] J. Kwac and R. Rajagopal. Demand response targeting using big data analytics. In *2013 IEEE International Conference on Big Data*, pages 683–690, 2013.

[31] J. Kwac, J. Flora, and R. Rajagopal. Lifestyle segmentation based on energy consumption data. *IEEE Transactions on Smart Grid*, 9(4):2409–2418, 2018.

[32] Omid Motlagh, Adam Berry, and Lachlan O'Neil. Clustering of residential electricity customers using load time series. *Applied energy*, 237:11–24, 2019.

[33] I Abubakar, SN Khalid, MW Mustafa, Hussain Shareef, and M Mustapha. Application of load monitoring in appliances' energy management–a review. *Renewable and Sustainable Energy Reviews*, 67:235–245, 2017.

[34] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[35] Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent dirichlet allocation. *advances in neural information processing systems*, 23:856–864, 2010.

[36] P. Pinoli, D. Chicco, and M. Masseroli. Latent dirichlet allocation based on gibbs sampling for gene function prediction. In *2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8, 2014.

[37] M. Lienou, H. Maitre, and M. Datcu. Semantic annotation of satellite images using latent dirichlet allocation. *IEEE Geoscience and Remote Sensing Letters*, 7(1):28–32, 2010.

[38] Leanne Giordono, Hilary Boudet, and Alexander Gard-Murray. Local adaptation policy responses to extreme weather events. *Policy sciences*, 53(4):609–636, 2020.

[39] Justin Johan Jozias van Der Hooft, Joe Wandy, Michael P Barrett, Karl EV Burgess, and Simon Rogers. Topic modeling for untargeted substructure exploration in metabolomics. *Proceedings of the National Academy of Sciences*, 113(48):13738–13743, 2016.

[40] Phil Grunewald and Marina Diakonova. The electricity footprint of household activities-implications for demand models. *Energy and Buildings*, 174:635–641, 2018.

[41] Erdal Aydin, Dirk Brounen, and Nils Kok. Information provision and energy consumption: Evidence from a field experiment. *Energy Economics*, 71:403–410, 2018.

[42] Peter Michael Gladhart, Bonnie Maas Morrison, and James J Zuiches. *Energy and families: Lifestyles and energy consumption in Lansing*. Michigan State University Press, 1987.

[43] Loren Lutzenhiser. Behavioral assumptions underlying california residential sector energy efficiency programs. *UC Berkeley: California Institute for Energy and Environment*, 2009.

[44] Jacopo Torriti. Understanding the timing of energy demand through time use data: Time of the day dependence of social practices. *Energy research & social science*, 25:37–47, 2017.

[45] Ben Anderson. Laundry, energy and time: Insights from 20 years of time-use diary data in the united kingdom. *Energy Research & Social Science*, 22:125–136, 2016.

[46] Alan Warde, Shu-Li Cheng, Wendy Olsen, and Dale Southerton. Changes in the practice of eating: A comparative analysis of time-use. *Acta Sociologica*, 50(4):363–385, 2007.

[47] Dale Southerton. Analysing the temporal organization of daily life: Social constraints, practices and their allocation. *Sociology*, 40(3):435–454, 2006.

[48] Trevor Memmott, Sanya Carley, Michelle Graff, and David M Konisky. Sociodemographic disparities in energy insecurity among low-income households before and during the covid-19 pandemic. *Nature Energy*, 6(2):186–193, 2021.

[49] Laura M Giurge, Ashley V Whillans, and Ayse Yemiscigil. A multicountry perspective on gender differences in time use during covid-19. *Proceedings of the National Academy of Sciences*, 118(12), 2021.

[50] Chad Zanocco, June Flora, Ram Rajagopal, and Hilary Boudet. Exploring the effects of california's covid-19 shelter-in-place order on household energy practices and intention to adopt smart home technologies. *Renewable and Sustainable Energy Reviews*, page 110578, 2020.

[51] Annika Todd, Peter Cappers, C Anna Spurlock, and Ling Jin. Spillover as a cause of bias in baseline evaluation methods for demand response programs. *Applied Energy*, 250:344–357, 2019.

[52] Robert H Socolow. The twin rivers program on energy conservation in housing: Highlights and conclusions. *Energy and Buildings*, 1(3):207–242, 1978.

[53] Martin Gerlach, Tiago P Peixoto, and Eduardo G Altmann. A network approach to topic models. *Science advances*, 4(7):eaaq1360, 2018.

[54] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[55] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

[56] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226. PMLR, 2015.

[57] Rishee K Jain, Junjie Qin, and Ram Rajagopal. Data-driven planning of distributed energy resources amidst socio-technical complexities. *Nature Energy*, 2(8):1–11, 2017.

[58] Holger Teichgraeber and Adam R Brandt. Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison. *Applied energy*, 239:1283–1293, 2019.

[59] Paul S Bradley, Olvi L Mangasarian, and W Nick Street. Clustering via concave minimization. *Advances in neural information processing systems*, pages 368–374, 1997.

[60] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[61] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[62] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[63] Sharon Xu, Edward Barbour, and Marta C González. Household segmentation by load shape and daily consumption. In *Proc. of. ACM SigKDD Workshop*, pages 1–9, 2017.

[64] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[65] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[66] Joana M Abreu, Francisco Câmara Pereira, and Paulo Ferrão. Using pattern recognition to identify habitual behavior in residential electricity consumption. *Energy and buildings*, 49:479–487, 2012.

[67] Alejandro Pena-Bello, Edward Barbour, MC Gonzalez, MK Patel, and David Parra. Optimized pv-coupled battery systems for combining applications: Impact of battery technology and geography. *Renewable and Sustainable Energy Reviews*, 112:978–990, 2019.

[68] Adrian Albert and Ram Rajagopal. Smart meter driven segmentation: What your consumption says about you. *IEEE Transactions on power systems*, 28(4):4019–4030, 2013.

[69] Juan Pablo Carvallo, Stephanie Bieler, Myles Collins, Joscha Mueller, Christoph Gehbauer, Douglas J Gotham, and Peter H Larsen. A framework to measure the technical, economic, and rate impacts of distributed solar, electric vehicles, and storage. *Applied Energy*, 297:117160, 2021.

[70] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

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

[72] Xiao Chen, Thomas Navidi, and Ram Rajagopal. Generating private data with user customization. *arXiv preprint arXiv:2012.01467*, 2020.

[73] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition, 2014.

[74] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[75] Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[76] Daniel Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

[77] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, 44(3):678–693, 2011.

[78] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 470–479. IEEE, 2014.

[79] Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. Ftw: fast similarity search under the time warping distance. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 326–337, 2005.

[80] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[81] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, pages 92–96, 2010.

[82] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120, 2006.

[83] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[85] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[86] 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.

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

[88] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[89] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[90] Martín Abadi and David G Andersen. Learning to protect communications with adversarial neural cryptography. *arXiv preprint arXiv:1610.06918*, 2016.

[91] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[92] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018.

[93] Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. *arXiv preprint arXiv:1812.04218*, 2018.

[94] Jihun Hamm. Minimax filter: learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734, 2017.

[95] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Deeprotect: Enabling inference-based access control on mobile sensing applications. *arXiv preprint arXiv:1702.06159*, 2017.

[96] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[97] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[98] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[99] Ilya Mironov. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

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

[101] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[102] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[103] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[104] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[105] Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn, and Wes B Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48:p411, 1994.

[106] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[107] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[108] Aria Rezaei, Chaowei Xiao, Jie Gao, and Bo Li. Protecting sensitive attributes via generative adversarial networks. *arXiv preprint arXiv:1812.10193*, 2018.

[109] Joonas Jälkö, Onur Dikmen, and Antti Honkela. Differentially private variational inference for non-conjugate models. *arXiv preprint arXiv:1610.08749*, 2016.

[110] Robert Torfason, Eirikur Agustsson, Rasmus Rothe, and Radu Timofte. From face images and attributes to attributes. In *Asian Conference on Computer Vision*, pages 313–329. Springer, 2016.

[111] Richard Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[112] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[113] Jiawei Chen, Janusz Konrad, and Prakash Ishwar. Vgan-based image representation learning for privacy-preserving facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1570–1579, 2018.

[114] Elliot Creager, David Madras, Joern-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. volume 97 of *Proceedings of Machine Learning Research*, pages 1436–1445, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[115] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[116] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

[117] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[118] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

[119] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[120] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

[121] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[122] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[123] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[124] Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

[125] Phillippe Rigollet and Jan-Christian Hütter. High dimensional statistics. *Lecture notes for course 18S997*, 2015.

[126] John Duchi, Khashayar Khosravi, Feng Ruan, et al. Multiclass classification, information, divergence and surrogate risk. *The Annals of Statistics*, 46(6B):3246–3275, 2018.

[127] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

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

[129] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial Intelligence and Statistics*, pages 277–286, 2015.

[130] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae:

Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[131] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 708–718, 2018.

[132] Alfred O Hero, Bing Ma, Olivier Michel, and John Gorman. Alpha-divergence for classification, indexing and retrieval. In *University of Michigan*. Citeseer, 2001.

[133] Andrey Bernstein and Emiliano Dall'Anese. Bi-level dynamic optimization with feedback. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, Canada, November 2017.

[134] Yuanyuan Shi, Bolun Xu, Di Wang, and Baosen Zhang. Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains. *IEEE Transactions on Power Systems*, 33:2882–2894, 2018.

[135] Thomas Navidi, Abbas El Gamal, and Ram Rajagopal. A two-layer decentralized control architecture for der coordination. In *2018 IEEE Conference on Decision and Control*, pages 6019–6024, 2018.

[136] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[137] Mikhail Lisovich, Deirdre Mulligan, and Stephen Wicker. Inferring personal information from demand-response systems. *Security & Privacy, IEEE*, 8:11–20, 01 2010.

[138] Marek Jawurek, Florian Kerschbaum, and George Danezis. Sok: Privacy technologies for smart grids–a survey of options. *Microsoft Res., Cambridge, UK*, 1:1–16, 2012.

[139] Nikos Komninos, Eleni Philippou, and Andreas Pitsillides. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials*, 16(4):1933–1954, 2014.

[140] Giulio Giaconi, Deniz Gunduz, and H Vincent Poor. Privacy-aware smart metering: Progress and challenges. *IEEE Signal Processing Magazine*, 35(6):59–78, 2018.

[141] Abhishek Halder, Xinbo Geng, PR Kumar, and Le Xie. Architecture and algorithms for privacy preserving thermal inertial load management by a load serving entity. *IEEE Transactions on Power Systems*, 32(4):3275–3286, 2016.

[142] Marwa Keshk, Elena Sitnikova, Nour Moustafa, Jiankun Hu, and Ibrahim Khalil. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. *IEEE Transactions on Sustainable Computing*, 2019.

[143] Niklas Buescher, Spyros Boukoros, Stefan Bauregger, and Stefan Katzenbeisser. Two is not enough: Privacy assessment of aggregation schemes in smart metering. *Proceedings on Privacy Enhancing Technologies*, 2017(4):198–214, 2017.

[144] Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 259–282, 2017.

[145] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[146] Lalitha Sankar, S Raj Rajagopalan, Soheil Mohajer, and H Vincent Poor. Smart meter privacy: A theoretical framework. *IEEE Transactions on Smart Grid*, 4(2):837–846, 2012.

[147] Shuo Han, Ufuk Topcu, and George J Pappas. Event-based information-theoretic privacy: A case study of smart meters. In *2016 American Control Conference (ACC)*, pages 2074–2079. IEEE, 2016.

[148] Jun-Xing Chin, Tomas Tinoco De Rubira, and Gabriela Hug. Privacy-protecting energy management unit through model-distribution predictive control. *IEEE Transactions on Smart Grid*, 8(6):3084–3093, 2017.

[149] Günther Eibl and Dominik Engel. Differential privacy for real smart metering data. *Computer Science-Research and Development*, 32(1-2):173–182, 2017.

[150] F. Zhou, J. Anderson, and S. H. Low. Differential privacy of aggregated dc optimal power flow data. In *2019 American Control Conference (ACC)*, pages 1307–1314, July 2019.

[151] Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. Differential privacy for power grid obfuscation. *IEEE Transactions on Smart Grid*, 2019.

[152] Dimitri P Bertsekas. Dynamic programming and optimal control 4th edition, volume ii. *Athena Scientific*, 2015.

[153] Mingxi Liu, Phillippe K Phanivong, and Duncan S Callaway. Customer-and network-aware decentralized ev charging control. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.

[154] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NIPS 2017 Workshop Autodiff Program*, 2017.

[155] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[156] Yize Chen, Xiyu Wang, and Baosen Zhang. An unsupervised deep learning approach for scenario forecasts. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.

[157] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 136–145. JMLR. org, 2017.

[158] Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M Moursi. Differentiating through a conic program. *Journal of Applied and Numerical Optimization*, 1:107–115, 2019.

[159] Yinyu Ye, Michael J Todd, and Shinji Mizuno. An o($\sqrt{n}L$)-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19(1):53–67, 1994.

[160] Enzo Busseti, Walaa M Moursi, and Stephen Boyd. Solution refinement at regular points of conic problems. *Computational Optimization and Applications*, pages 1–17, 2018.

[161] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[162] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[163] Commission for Energy Regulation. Smart metering project-electricity customer behaviour trial, 2009-2010. http://www.ucd.ie/issda/data/commissionforenergyregulationcer/, 2012.