Thèse n° 9523

Challenging the Assumptions: Rethinking Privacy, Bias, and Security in Machine Learning

Présentée le 20 octobre 2023

Faculté informatique et communications
Laboratoire d’ingénierie de sécurité et privacy
Programme doctoral en informatique et communications

pour l’obtention du grade de Docteur ès Sciences

par

Bogdan KULYNCH

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. C. González Troncoso, directrice de thèse
Prof. A. Oprea, rapporteuse
Prof. B. Ustun, rapporteur
Prof. N. Flammarion, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

Acknowledgements

We often cheekily rephrase the famous proverb: *It takes a village to finish a Ph.D.* This statement is undoubtedly true. In the following, I am making my best effort to thank those whose mentorship and support have made this thesis possible, although this effort is unavoidably incomplete.

First, I want to thank my Ph.D. advisor Carmela Troncoso. Carmela took a chance on me some seven years ago and offered unwavering support throughout all this time. I am incredibly fortunate to have such a compassionate advisor that has had to withstand my stubbornness and pathological multi-tasking yet still has given me the freedom to follow my pursuits. Carmela has been there for me, both in academic matters as well as personal challenges. I have learned a lot from her expertise, but something especially valuable she has shown me is the importance of never compromising on your values. This lesson is something I will certainly carry into the future.

I want to thank Flavio du Pin Calmon for hosting me for a short but very fulfilling semester in Boston. Flavio's guidance has enabled me to look deeper into many aspects of my research topics. Something that I genuinely appreciate about Flavio is his contagious optimism and passion for teaching and science. My inner skeptic has given in to this contagion and decided to keep—maybe just a little bit—of that optimism.

I am grateful to many other colleagues who have provided support, guidance, and mentorship on this path. In particular, I want to thank Seda Gürses for all the new perspectives I learned in our collaborations and for giving me a chance to participate in multiple exciting events, and my long-time officemate Wouter Lueks for all the help and advice, as well as for many hours of small and not so small talk. I also want to specifically acknowledge all the teachers at the Kyiv Mohyla Academy, without whom I would not have gotten to the point of doing a Ph.D. in the first place. In particular, these are Volodymyr Boublik and Andrii Glybovets, who showed me and countless others that Computer Science is exciting, and Bogdana Oliynyk, who has encouraged me to pursue an academic career. Finally, I also want to acknowledge my school teachers at Kyiv's School #172: Natalia Leonenko, who instilled an interest in science in me; Natalia Starodub, without whom I would not have been writing this in English; and Olena Kurylina, without whom I would not have been writing anything.

I am incredibly grateful to all the fantastic collaborators I have had the pleasure to work with throughout these years. I would like to particularly thank Rebekah Overdorf for her insights, positive attitude, and company; Deborah Raji, Jacob Metcalf, Manny Moss, Andrew Smart, and Kristian Lum for the conversations I have always been

looking forward to and that have always become the highlight of my day; Preetum Nakkiran who showed a true master-class in effectively turning an idea into reality; Kendra Albert, Ram Shankar Siva Kumar, Maggie Delano, Piotr Sapieżyński, Michael Veale, Jamie Hayes, and Agathe Balayn for deep insights across various disciplines.

During my Ph.D., I have had the incredible privilege of being in the best possible environments. EPFL is home to some of the most excellent researchers I have met, and I am happy that I have had a chance to work together, workshop ideas and talks, and sometimes even take a break and relax with these folks. They include Giovanni Cherubin, Mohammad Yaghini, Sandra Siby, Theresa Stadler, Kasra Edalatnejadkhamene, Marion Le Tilly, Laurent Girod, Mathilde Raynal, Klim Kireev, Boya Wang, Sylvain Chatel, Christian Mouchet, Sinem Sav, Dario Pasquini, Vera Rimmer at the SPRING Lab, as well as Ceyhun Alp, Kirill Nikitin, Ludovic Barman, David Froelicher, Linus Gasser, Vijay Keswani, Michał Tyszkiewicz, Jakab Tardos, and Maksym Andriushchenko from the other corners of the university. I am happy to have had the most exciting and mathematically involved discussions over cups of coffee and at the whiteboards with Juan Felipe Gomez, Hsiang Hsu, Carol Long, Wael Alghamdi, Lucas Monteiro Paes, Hao Wang, and Alex Oesterling at Harvard.

Finally, I want to thank my jury, Alina Oprea, Berk Ustun, Nicolas Flammarion, and Pascal Frossard, for their feedback; the colleagues who have helped with thesis proof-reading and translations: Maksym Andriushchenko, Sylvain Chatel, and Mathilde Raynal; and Isabelle Coke for invaluable help with all the bureaucratic processes.

These years would have been much bleaker without dear friends. In Lausanne, I want to thank Sandra and Karen for being the most cheerful, understanding, and supportive friends one could hope for; Bharath for the thought-provoking reflections through his socratic dialogs interspersed with bad jokes and giggles; Jakab, Bernd, and Maksym, all of whom are somehow capable of reasoning about technical subjects after running up a two-kilometer high mountain, for their company; Elene, for the bursts of creativity; Theresa for the chats and our lake outings; as well as Merlin, Sena, Sahand, Nastia, Jojo, Matteo, Krzysztof, Stephen and many others for numerous joyous moments. In Boston, I am so happy to have had a chance to meet fantastic Zana, Yevhenii, Viktoriia, Hsiang, Arcady, Tanya, and Piotr. Our conversations have certainly been transformative for me. I want to thank Natalia, Viona, Pattabi, and Arsen for being wonderful hosts in California. I am grateful for the company to all the folks at IMDEA in Madrid, particularly Borja, Elena, and Srdjan; to Marios, Alex, Sarah, and Maciek in London and other places where we have met; and to Sasha S., Misha, and Sasha C. for the pandemic Zoom chats. In Kyiv, I want to thank friends who have supported me, particularly Viktor, who has done so through his intellectual duels, and Yaroslav, through his easy-going attitude. I would like to especially acknowledge my dear friends Pavlo K., the most passionate person when it comes to building resilient communities and comfortable urban environments, and Sasha, the most curious and creative technologist I know, who have both enlisted and now serve in Ukraine's Armed Forces; and Pavlo B., the most principled individual I know, who now spends his days and nights working on the vital army supplies during the ongoing genocidal invasion.

I am grateful to my parents for providing a safety net and doing everything possible so that I could get an education. In a nostalgic throwback to my youth, my parents, sister, and little nephew have kept me company in Lausanne during the last year of my Ph.D. Despite challenging circumstances, we shared many memorable moments over some syrnyky, varenyky, and holubtsi. Last but not least, no words can express my gratitude to Mariam, my closest friend, partner, and beloved. Her support, unique perspectives, perseverance, curiosity, and humor have been a source of strength for me throughout the numerous ups and downs during this time. She moved countries and changed careers to make both of our lives better. I would not be the same person, and I would not have been able to pull this off without her.

Lausanne, August 5, 2023 Bogdan Kulynych

Abstract

Predictive models based on machine learning (ML) offer a compelling promise: bringing clarity and structure to complex natural and social environments. However, the use of ML poses substantial risks related to the privacy of their training data as well as the security and reliability of their operation. This thesis explores the relationships between privacy, security, and reliability risks of ML. Our research aims to re-evaluate the standard practices and approaches to mitigating and measuring these risks in order to understand their connections and scrutinize their effectiveness.

The first area we study is data privacy, particularly the standard privacy-preserving learning technique of differentially private (DP) training. DP training introduces controlled randomization to limit information leakage. This randomization has side effects such as performance loss and widening of performance disparities across population groups. In the thesis, we investigate additional side effects. On the positive side, we highlight the "What You See Is What You Get" property that DP training achieves. Models trained with standard methods often exhibit significant differences between training and testing phases, whereas privacy-preserving training guarantees similar behavior. Leveraging this property, we introduce competitive algorithms for group-distributionally robust optimization, addressing privacy-performance trade-offs, and mitigating robust overfitting. On the negative side, we show that decisions of DP-trained models can be arbitrary: due to the randomness in training, equally private models can yield drastically different predictions for the same input. We examine the costs of standard DP training algorithms in terms of arbitrariness, raising concerns about the justifiability of their decisions in high-stakes scenarios.

Next, we study the standard measure of privacy leakage: vulnerability of models to membership inference attacks. We analyze how the vulnerability to these attacks, thus privacy risks, are unequally distributed across the population groups. We emphasize the need and provide methods to consider privacy leakage across diverse subpopulations to avoid disproportionate harm and address inequities.

Finally, our study focuses on analyzing the security risks in tabular domains, which are commonly found in high-stakes ML settings. We challenge the assumptions behind existing security evaluation methods, which primarily consider threat models based on input geometry. We highlight that real-world adversaries in these settings face practical constraints, prompting the need for cost and utility-aware threat models. We propose a framework that tailors adversarial models to tabular domains, enabling the consideration of cost and utility constraints in high-stakes decision-making situations.

Overall, the thesis sheds light on the subtle effects of DP training, emphasizes the importance of diverse subpopulations in risk measurements, and highlights the need for realistic threat models and security measures. By challenging assumptions and re-evaluating risk mitigation and measurement approaches, the thesis paves the way for more robust and ethically grounded studies of ML risks.

Keywords: machine learning, differential privacy, membership inference attacks, algorithmic fairness, predictive multiplicity, adversarial robustness, adversarial examples

Résumé

Les modèles prédictifs basés sur l'apprentissage automatique (machine learning ; ML) offrent une promesse convaincante : apporter clarté et structure à des environnements naturels et sociaux complexes. Cependant, l'utilisation de l'apprentissage automatique pose des risques substantiels liés à la confidentialité des données d'apprentissage ainsi qu'à la sécurité et à la fiabilité de leur fonctionnement. Cette thèse explore les relations entre les risques liés à la confidentialité, à la sécurité et à la fiabilité. Notre recherche vise à réévaluer les pratiques et les approches standard pour atténuer et mesurer ces risques afin de comprendre leurs liens et d'examiner leur efficacité.

Le premier domaine que nous étudions est celui de la confidentialité des données, en particulier la technique d'apprentissage classique préservant la confidentialité, à savoir l'entraînement différentiellement privé (DP). L'entraînement DP introduit une randomisation contrôlée pour limiter les fuites d'informations. Cette randomisation a des effets secondaires tels que la perte de performances et l'augmentation des disparités de performance entre les groupes de population. Dans cette thèse, nous étudions d'autres effets secondaires. Du côté positif, nous soulignons la propriété "Ce que vous voyez est ce que vous obtenez" que l'entraînement DP permet d'obtenir. Les modèles formés avec des méthodes standard présentent souvent des différences significatives entre les phases d'entraînement et de test, alors que l'entraînement préservant la confidentialité garantit un comportement similaire. En tirant parti de cette propriété, nous introduisons des algorithmes compétitifs pour l'optimisation distributionnelle de groupe, en abordant les compromis entre confidentialité et performance, et en atténuant le surajustement robuste. Du côté négatif, nous montrons que les décisions des modèles entraînés par DP peuvent être arbitraires : en raison du caractère aléatoire de l'entraînement, des modèles également privés peuvent produire des prédictions radicalement différentes pour la même entrée. Nous examinons les coûts des algorithmes standard d'entraînement DP vis-à-vis de ces décisions arbitraires, ce qui soulève des inquiétudes quant à la justification de leurs décisions dans les scénarios à fort enjeu.

Ensuite, nous étudions la mesure classique de l'atteinte à la vie privée : la vulnérabilité des modèles aux attaques par inférence d'appartenance. Nous analysons comment la vulnérabilité à ces attaques, et donc les risques d'atteinte à la vie privée, sont inégalement répartis entre les groupes de population. Nous insistons sur la nécessité de prendre en compte la fuite de données personnelles dans diverses sous-populations afin d'éviter des dommages disproportionnés et de remédier aux inégalités.

Enfin, notre étude se concentre sur l'analyse des risques de sécurité dans les domaines tabulaires, que l'on trouve couramment dans les contextes de ML à enjeu élevé. Nous remettons en question les hypothèses sous-jacentes des méthodes d'évaluation de la sécurité existantes, qui considèrent principalement les modèles de menace basés sur la géométrie d'entrée. Nous soulignons que les adversaires du monde réel, dans ces contextes, sont confrontés à des contraintes pratiques, d'où la nécessité de considerer des modèles de menace tenant compte des coûts et de l'utilité. Nous proposons un cadre qui adapte les modèles d'adversaires aux domaines tabulaires, ce qui permet de prendre en compte les contraintes de coût et d'utilité dans les situations de prise de décision à fort enjeu.

En conclusion, notre thèse met en lumière les effets subtils de l'entraînement DP, souligne l'importance de la diversité des sous-populations dans la mesure des risques et met en évidence la nécessité de modèles de menace et de mesures de sécurité réalistes. En remettant en question les hypothèses et en réévaluant les approches d'atténuation et de mesure des risques, la thèse ouvre la voie à des études plus robustes et plus éthiques, fondées sur les risques liés à l'apprentissage automatique.

Mots clés : apprentissage automatique, confidentialité différentielle, attaques d'inférence d'appartenance, équité algorithmique, multiplicité prédictive, robustesse aux adversaires, exemples adversariaux.

Contents

List of Figures

List of Tables

Introduction

In a time defined by the increasing reliance on data-driven decision-making, predictive models based on machine learning (ML) present an especially compelling promise. This promise is to bring legibility [4, 155]—clarity and structure—to the unclear and unstructured mess of natural and social environments. Legibility, in turn, is thought to enable companies, organizations, and governments to optimize, control, and rule those environments effectively. The use of ML, however, comes with risks.

A crucial outcome of ML models being built on data is that the resulting predictive models might give a false impression of being an objective tool that provides an unbiased gaze from nowhere. In practice, they are anything but unbiased. Being trained on historical data, they are bound to reproduce both the good and the ugly sides of the processes that generated the data [11]. Moreover, such predictive models could issue predictions that are influenced by either their designers' arbitrary decisions or even the randomness involved in their creation or operation [18]. This influence is especially concerning in high-stakes settings such as healthcare and scoring of credit or fraud risk. In these settings, the combination of the false promise of objectivity and unbiasedness with the arbitrariness of decisions is bound to create bureaucratic-technological traps [4, 39] for the decision subjects of these models. Even more, as the predictions can be arbitrary to a degree, the predictive models can also be gamed into producing arbitrary outputs through simple manipulations of their inputs [71, 77].

Another inherent risk resulting from ML being built on data comes when the data contains privacy-sensitive information. It might not be evident to practitioners that the ML models can leak this information when no precautions are in place [161], even if the data itself is collected and stored securely. The European Union data protection regulation (GDPR), as well as similar legislation around the world [73], mandates strict data protection, which, arguably, covers such data leakage from trained models [172].

This thesis makes a step towards understanding these seemingly different yet closely interconnected privacy, security, and reliability risks within a broader social and technical context. Instead of studying them in isolation, we aim to re-evaluate the standard practices and approaches to mitigating and measuring these risks considering their interactions and intersections. What are some inherent relationships between the risks? Do mitigations of these risks hold up to scrutiny within such a broader view? We summarize our contributions toward answering these questions next.

Side Effects of Differential Privacy. We begin our exploration from the perspective of data privacy. To address the previously mentioned concerns with privacy leakage, scholars have proposed multiple techniques for privacy-preserving learning, which aim to safeguard sensitive data while enabling effective model training. In particular, we study a standard way to learn while ensuring privacy, differentially private (DP) training. This approach relies on a somewhat disruptive intervention to the training process of ML models: introducing a controlled level of randomization. Random noise injected as part of the training process limits the potential leakage of information from the data but comes with side effects. Some well-known side effects are the loss of performance on average and the widening of performance disparities across population groups. In this thesis, we aim to explore the additional effects that can arise from incorporating random noise during the training process. We delve into this question in Chapters 2 and 3.

In *Chapter 2,* we study a positive side effect of DP training beyond privacy. Privacy-preserving training achieves similar behavior between training and deployment (test) time, which we refer to as the "What You See Is What You Get" (WYSIWYG) property. This is in contrast to models trained with standard non-private methods whose behaviors can differ significantly between the training and testing phases. We show this by quantifying this property with a notion of distributional generalization, a measure of the similarity of outputs of the predictive models between training and test data. Leveraging this connection, we construct simple algorithms that outperform state-of-the-art approaches to train models for robust performance across population groups, improve the trade-offs between privacy and performance disparities, and mitigate artifacts of securing models against input manipulations. Thus, the randomization introduced in DP training can have unexpected—in a good sense—side effects spanning beyond privacy and standard on-average performance measures.

In *Chapter 3,* we turn to a negative side effect of DP training. We reveal a significant but unnoticed effect of randomization in training: it leads to the arbitrariness of decisions. Specifically, equally-private models can provide drastically different predictions for the same input example due to randomness in training. We investigate the costs of several algorithms for DP training in terms of arbitrary decisions, both theoretically and through extensive experiments. Our findings show that as the level of privacy increases, the degree of arbitrariness invariably rises across tasks and models, impacting different individuals and demographic groups differently. This raises concerns about the justifiability of decisions made using this kind of privacy-preserving models in high-stakes scenarios.

Unequal Access to Privacy. Next, we take a step back and focus on the methods to measure privacy leakage. In *Chapter 4,* we study the standard measure of privacy leakage: vulnerability to membership inference attacks. Membership inference attacks are a fundamental threat to the privacy of the training data in which an attacker aims to guess whether a given data record was used as part of the training data or not. Vulnerability to these attacks is a manifestation of data leakage. Our first finding is that a standard notion of vulnerability to membership inference is equivalent to the notion of distributional generalization described previously. This result bridges the concepts of performance quality and privacy in ML. We use this finding as a basic theoretical tool for investigating the fundamental issue with the standard approaches to measuring leakage: privacy leakage is not adequately studied across diverse subpopulations. We find that privacy risks can vary significantly across different groups. Ignoring this disparate impact and failing to account for the uneven distribution of privacy leakage could result in disproportionate harm and exacerbate existing inequities.

Realistic Adversarial Modeling for Tabular Data. Having explored the intersections of privacy on the one hand and bias and arbitrariness on the other, we turn to the security risks in the final part of the thesis. In *Chapter 5,* we show that the current landscape of attacks against ML models primarily revolves around threat models that focus on the geometry of inputs. In reality, adversaries face practical limitations such as cost and utility constraints, making the real-world threat landscape significantly different from the idealized scenarios considered in the existing attack methodologies. This discrepancy between the assumptions made in attack models and the realistic constraints faced by adversaries raises concerns about the effectiveness and real-world applicability of current security measures in ML. To address these limitations, we propose cost and utility-aware threat models tailored explicitly to attackers targeting tabular domains, in which the decisions of the predictive models are often high-stakes. Our framework enables the design of attack mechanisms that take into account cost and utility constraints, such as a financial budget. We demonstrate the effectiveness of our approach on realistic ML tasks with economic and social implications.

In sum, we highlight the nuanced effects of differentially private training on model behavior beyond privacy, the importance of considering diverse subpopulations in privacy measurements, and the need for realistic threat models and security measures. While delving into these topics, we have challenged and re-evaluated assumptions in the mitigation and measurement of privacy, security, and reliability risks of ML. We have provided tools that enable practitioners and regulators to have a broad picture of these risks, paving the way for more robust and ethically grounded approaches to studying, evaluating, and mitigating them.

Chapter 1

Technical Preliminaries

In this chapter, we briefly introduce concepts and notation that commonly occur throughout the thesis.

Common notation. We use I_d to denote the d-by-d identity matrix, $\mathbb{1}[\cdot]$ to denote the indicator function, and 2^V to denote the power set of the set V. We use L_p to refer to a p-norm, defined for $0 < p < \infty$ and an input $x \in \mathbb{R}^d$ as follows:

$$\|x\|_p \triangleq \left(\sum_{i=1}^{d} |x_i|^p\right)^{\frac{1}{p}}. \tag{1.1}$$

For a special case of $p = \infty$, it is defined as:

$$\|x\|_\infty \triangleq \max_{i=1,\ldots,d} |x_i|. \tag{1.2}$$

An L_p norm for $p < \infty$ can have a weighted variant. For a vector $w \in \mathbb{R}^d$, we define a w-weighted L_p norm as follows:

$$\|x\|_{p,w} = \left(\sum_{i=1}^{d} w_i \cdot |x_i|^p\right)^{\frac{1}{p}}. \tag{1.3}$$

1.1 Probability, Statistics, and Statistical Distance

We denote by $\mathcal{N}(\mu, \Sigma)$ the d-dimensional normal distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. If $\Sigma = \sigma^2 \cdot I_d$, then we call the distribution isotropic. We denote by $\Phi(\cdot)$ the cumulative distribution function of a standard normal distribution.

Probability distributions. We commonly make use of analyses based on the theory of probability and statistics. We study *probability distributions* P over a given *sample space* $\mathbb{D}$. We denote a *random variable* X over the sample space $\mathbb{D}$ that is distributed according to the distribution P as $X \sim P$. For any subset of the sample space $V \subseteq \mathbb{D}$, the distribution defines a function which satisfies:

$$P(V) = \Pr[X \in V]. \tag{1.4}$$

If V is a finite set, in a slight abuse of notation we also use $X \sim V$ to denote that X is a random variable uniformly distributed on V.

For a given probability distribution P over $\mathbb{D}$ and a function $\pi : \mathbb{D} \to \mathbb{R}^d$, we use $\pi \sharp P$ to denote a *pushforward*, the distribution of $\pi(X)$ for $X \sim P$.

A *continuous* probability distribution P over $\mathbb{R}$ is the one that has an associated *probability density function* $v : \mathbb{R} \to \mathbb{R}$ such that for any $V \subseteq \mathbb{R}$:

$$P(V) = \int_V v(x) \,\mathrm{d}x \tag{1.5}$$

Statistical distance. Suppose that P and Q are two probability distributions over a space $\mathbb{D}$. *Statistical distance*, also known as the *total-variation (TV) distance* is a measure of difference between the two distributions, defined as follows:

$$d_{\mathsf{TV}}(P, Q) \triangleq \sup_{V \subseteq \mathbb{D}} |P(V) - Q(V)|. \tag{1.6}$$

Next, we highlight several useful properties of the TV distance. An important property is that it is non-increasing under *post-processing*: For any *post-processing function* $\pi : \mathbb{D} \to \mathbb{R}^k$, it holds that:

$$d_{\mathsf{TV}}(\pi \sharp P, \pi \sharp Q) \leq d_{\mathsf{TV}}(P, Q). \tag{1.7}$$

TV distance is a metric. In particular, it satisfies the triangle inequality:

$$d_{\mathsf{TV}}(P, Q) \leq d_{\mathsf{TV}}(P, W) + d_{\mathsf{TV}}(W, Q), \tag{1.8}$$

where P, Q, W are any probability distributions on the same space $\mathbb{D}$.

The total-variation distance takes on the following useful forms:

- It has the equivalent *variational form*:

$$d_{\mathsf{TV}}(P, Q) = \sup_{\phi : \mathbb{D} \to [0,1]} \left| \mathop{\mathbb{E}}_{X \sim P} \phi(X) - \mathop{\mathbb{E}}_{Y \sim Q} \phi(Y) \right| \tag{1.9}$$

- Over a binary sample space $\mathbb{D} = \{0, 1\}$, the TV distance simplifies to the difference of expectations:

$$d_{\mathsf{TV}}(P, Q) = \left| \mathop{\mathbb{E}}_{X \sim P}[X] - \mathop{\mathbb{E}}_{Y \sim Q}[Y] \right| \tag{1.10}$$

- In the case that P and Q are both continuous probability distributions over $\mathbb{R}$ with density functions $v_P(x)$ and $v_Q(x)$, respectively, the total-variation distance can be written as:

$$d_{\mathsf{TV}}(P, Q) = \frac{1}{2} \int_{\mathbb{R}} |v_P(x) - v_Q(x)| \, \mathrm{d}x. \tag{1.11}$$

We refer to Polyanskiy and Wu [144] for an in-depth treatment of the subject.

1.2 Learning to Predict

Statistical learning. Throughout the thesis, we consider a *machine learning (ML) task* with a *population* of labeled *examples* $\mathbb{D} \triangleq \mathbb{X} \times \mathbb{Y}$, where $\mathbb{X}$ is the space of examples (also referred to as *feature vectors*) and $\mathbb{Y}$ is the space of labels. The goal of the learning task is to produce a *predictive model* that, given an example and its label $z \triangleq (x, y)$ can predict the label $y \in \mathbb{Y}$ by only observing the feature vector $x \in \mathbb{X}$. Given a *dataset* $S \in \mathbb{D}^n$, we use a possibly randomized *training algorithm* $T : \mathbb{D}^n \to \Theta$ that outputs a parameter vector θ of a predictive model from the set Θ. The predictive model parameterized by θ defines a prediction function $f_\theta(x)$, which aims to reproduce the label y of a given example $z = (x, y)$.

We capture the error of the prediction using a *loss function* $\ell(z; \theta) > 0$, with higher values of the loss function representing a higher degree of error. In *classification tasks*, that is, settings where $\mathbb{Y}$ is discrete and finite, a loss function that we commonly use in our analyses is the *0-1 loss*: $\ell((x, y); \theta) = \mathbb{1}[f_\theta(x) \neq y]$.

For some of the analyses in the thesis, we assume that there exists a *data distribution* of labeled examples P defined over the data space $\mathbb{D} = \mathbb{X} \times \mathbb{Y}$. We denote sampling of a labeled example $z = (x, y)$ from this distribution as $z \sim P$. We then further make a standard assumption that a dataset $S \in \mathbb{D}^n$ that is input to the training algorithm is an independently and identically distributed (i.i.d.) sample from P, denoted as $S \sim P^n$.

In this probabilistic setting, a common measure of the quality of a predictive model is its *expected loss* (equivalently, *expected error*, or *expected risk*):

$$R(\theta) \triangleq \mathop{\mathbb{E}}_{z \sim P}[\ell(z; \theta)]. \tag{1.12}$$

We are often interested in finding predictive models that minimize the expected loss.

Binary classification. In some parts of the thesis, we focus on the setting of binary classification in which there are only two possible labels, $\mathbb{Y} = \{0, 1\}$. In this case, we assume that the predictive model has a special structure. The model (*classifier* in this case) associates a *confidence score* to each input $x \in \mathbb{X}$, denoted as $h_\theta(x) \in [0, 1]$. If the confidence score is higher than some threshold $\tau \in [0, 1]$, then the decision is *positive* ($y = 1$). Otherwise, it is *negative* ($y = 0$). The classifier's prediction is thus obtained by applying a threshold to the confidence score:

$$f_\theta(x) \triangleq \mathbb{1}[h_\theta(x) > \tau]. \tag{1.13}$$

In the rest of the thesis, we assume a standard threshold of $\tau = 0.5$.

Bayes error. Under the probabilistic assumption that the data comes from a distribution P, the classifier that achieves the minimum possible expected loss is called the *Bayes (or Bayes-optimal) classifier*. In the case of the 0-1 loss, the Bayes classifier has a closed form:

$$f^*(x) \triangleq \max_{y \in \{0,1\}} \Pr_{(x,y) \sim P}[y \mid x]. \tag{1.14}$$

The expected error of the Bayes classifier is called the *Bayes error*. For our case of 0-1 loss, it is defined as:

$$R^* \triangleq \mathop{\mathbb{E}}_{(x,y) \sim P} \mathbb{1}[f^*(x) \neq y] = \Pr[f^*(x) \neq y] \tag{1.15}$$

By definition, the Bayes error is at least as low as the error of any other possible classifier in terms of the same loss function. That is, for any parameter vector $\theta \in \Theta$ we have:

$$R^* \leq R(\theta) = \mathop{\mathbb{E}}_{(x,y) \sim P} \mathbb{1}[f_\theta(x) \neq y] = \Pr[f_\theta(x) \neq y] \tag{1.16}$$

The Bayes error has a useful characterization in the case of *balanced* binary classification, that is, when the marginal probabilities of classes are equal: $\Pr[y = 1] = \Pr[y = 0] = {}^1\!/_2$. This setting is also known as having the *uniform prior*.

Consider the *class-conditional probability distributions* P_1 and P_0, defined for any $V \subseteq \mathbb{X}$

Algorithm 1 SGD

```
Input: Dataset $S$, loss function $\ell(z;\theta)$, initial parameters $\theta_0$, learning rate $\eta$, number of steps $t_{\max}$, mini-batch size $b$
Output: Parameters of the trained model $\theta_{t_{\max}}$
  for $t = 1, \dots, t_{\max}$ do
      Sample mini-batch $S_t \leftarrow \mathsf{Sample}_b(S)$
      $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \frac{1}{b} \sum_{z \in S_t} \nabla \ell(z; \theta_{t-1})$
```

as follows:

$$\begin{aligned} P_1(V) &\triangleq \Pr[x \in V \mid y = 1] \\ P_0(V) &\triangleq \Pr[x \in V \mid y = 0], \end{aligned} \tag{1.17}$$

over the randomness of $(x, y) \sim P$. Then, the Bayes error is proportional to the TV distance between the class-conditional distributions:

$$R^* = \mathop{\mathbb{E}}_{(x,y)\sim P} \mathbb{1}[f^*(x) \neq y] = {}^1\!/_2 - {}^1\!/_2 \cdot d_{\mathsf{TV}}(P_1, P_0) \tag{1.18}$$

We refer to Devroye et al. [49] for additional details.

Empirical risk minimization. A standard approach for obtaining the parameters of a predictive model tailored to a given dataset S is finding optimal parameters θ^* that minimize *empirical risk*—the average loss over the examples in the dataset—as follows:

$$\theta^* \in \arg\min_\theta \frac{1}{n} \sum_{z \in S} \ell(z; \theta). \tag{1.19}$$

As long as the dataset size is large enough, empirical risk approximates the expected error. There are different ways to solve this optimization problem depending on the type of the predictive model. We present one of the algorithms to do so next.

Stochastic gradient descent. A common method for training predictive models that are differentiable in their parameters θ is *stochastic gradient descent* (SGD), presented in Algorithm 1. SGD also serves as a basic building block for more complex training algorithms. Given a dataset S and an initial vector of parameters θ, the algorithm randomly samples multiple *mini-batches* of size $b \leq |S|$. We denote the sampling procedure which returns a mini-batch by $\mathsf{Sample}_b(S)$. For each mini-batch, the algorithm then computes the gradient of the (differentiable) loss function $\nabla_\theta \ell(z; \theta)$ for each z in the mini-batch, and updates the parameters by the scaled average gradient.

Generalization and overfitting. *Generalization* is a measure of closeness between the empirical error of the model θ on the training dataset S to its expected error:

$$\left|\frac{1}{n}\sum_{z\in S}\ell(z;\theta) - \mathop{\mathbb{E}}_{z\sim P}[\ell(z;\theta)]\right| \tag{1.20}$$

When the gap in Eq. (1.20) is large—where the definition of large depends on the concrete learning setting—we say that the model θ *overfits* to its training data.

In practice, generalization is approximated using an independent dataset sample $\bar{S} \sim P^m$, called the *test dataset* as follows:

$$\left|\frac{1}{n}\sum_{z\in S}\ell(z;\theta) - \frac{1}{m}\sum_{z\in \bar{S}}\ell(z;\theta)\right| \tag{1.21}$$

We also use a related notion of generalization, called *on-average generalization*, or *generalization in expectation* [159]:

$$\left|\mathop{\mathbb{E}}_{\substack{S\sim P^n\\ z\sim S}}\ell(z;T(S)) - \mathop{\mathbb{E}}_{\substack{S\sim P^n\\ z\sim P}}\ell(z;T(S))\right|, \tag{1.22}$$

where the first term is a shorthand notation:

$$\mathop{\mathbb{E}}_{\substack{S\sim P^n\\ z\sim S}}\ell(z;T(S)) \triangleq \mathop{\mathbb{E}}_{S\sim P^n}\frac{1}{n}\sum_{z\in S}\ell(z;T(S)). \tag{1.23}$$

Population subgroups. The data space could contain distinct groups (or subgroups) of examples. We use the words *group* and *subgroup* interchangeably. In the case that the data describes people, such groups could represent salient demographic populations such as those corresponding to a gender, race, ethnicity, etc. Formally, we assume the data distribution P is a mixture of m groups indexed by the set $\mathbb{G} = \{G_1, \ldots, G_m\}$, such that $P = \sum_{i=1}^m q_i P_i$. The vector $(q_i, \ldots, q_m) \in [0,1]^m$ represents the group probabilities, with $\sum_{i=1}^m q_i = 1$. We denote a group as $G \in \mathbb{G}$ and its corresponding distribution as P_G.

We make the following assumption about the relationship between the example and its group: for any $x \in \mathbb{X}$ we can determine the group to which x belongs as $G = g(x)$. In a slight abuse of notation, we also write $g(z)$ to denote the group of a labeled example. This assumption commonly holds in practice, e.g., when a group annotation G is a part of the feature vector x. For a dataset S, we denote its part consisting only of examples belonging to group $G \in \mathbb{G}$ as S_G.

1.3 Differentially Private Learning

In this thesis, we often consider learning tasks where the training data is privacy-sensitive (e.g., healthcare). In such settings, *learning with differential privacy (DP)* is one of the standard approaches to train predictive models [54, 56]. A randomized learning algorithm $T : \mathbb{D}^n \rightarrow \Theta$ is (ϵ, δ)-differentially private (DP) if for any two *neighbouring datasets* (i.e., datasets differing by at most one example) S, S', for any subset of parameter vectors $V \subseteq \Theta$, it holds that

$$\Pr[T(S) \in V] \leq \exp(\epsilon) \Pr[T(S') \in V] + \delta. \tag{1.24}$$

We denote the fact that two datasets are neighbouring as $S \simeq S'$. Informally, the respective probability distributions of models produced on any two neighbouring datasets should be similar to a degree defined by parameters (ϵ, δ). The parameters represent the level of privacy: low ϵ and low δ mean better privacy. DP mathematically encodes a notion of plausible deniability of the inclusion of an example in the dataset.

A special case when $\delta = 0$ is called *pure DP*. In this case, we denote that an algorithm satisfies the condition by omitting $\delta = 0$ and simply writing it as ϵ-DP.

The definition can be thought as an upper bound on a special probability distance between the distributions of $T(S)$ and $T(S')$. As is the case with TV distance, the distance constrained by DP is also non-increasing under post-processing. Thus, if an algorithm $T : \mathbb{D}^n \rightarrow \Theta$ satisfies (ϵ, δ)-DP, then for any post-processing function $\pi : \Theta \rightarrow \mathbb{R}^k$, their composition $\pi(T(S))$ also satisfies (ϵ, δ)-DP.

There is a multitude of methods that achieve DP. Next, we present two important ones.

Output perturbation. Output perturbation [33, 148, 190] is a simple method for achieving DP that takes an output parameter vector of a non-private training procedure, and privatizes it by adding random noise, e.g., sampled from the isotropic Gaussian distribution. Concretely, suppose that $T_{\mathsf{np}} : \mathbb{D}^n \rightarrow \Theta$ is a non-private learning algorithm. Denoting its output parameters as $\theta_{\mathsf{np}} = T_{\mathsf{np}}(S)$, we obtain the privatized parameters $\theta_{\mathsf{priv}} \in \Theta$ as:

$$\theta_{\mathsf{priv}} = \theta_{\mathsf{np}} + \xi, \text{ where } \xi \sim \mathcal{N}(0, \sigma^2 I_d). \tag{1.25}$$

The exact level of DP provided by this procedure depends on the specifics of the non-private training algorithm $T_{\mathsf{np}}(S)$. In particular, in order to achieve (ϵ, δ)-DP, it is sufficient to set the noise scale as follows:

$$\sigma = C \cdot \frac{\sqrt{2 \log(1.25/\delta)}}{\epsilon}, \tag{1.26}$$

where $C \triangleq \max_{S \simeq S'} \|T_{\text{np}}(S) - T_{\text{np}}(S')\|_2$ is the *sensitivity* of the non-private training

Algorithm 2 DP-SGD

```
Input: Dataset $S$, loss function $\ell(z;\theta)$, initial parameters $\theta_0$, learning rate $\eta$, maximal gradient norm $C$, noise parameter $\sigma$, number of steps $t_{\max}$, sampling rate $p$
Output: Parameters of the trained model $\theta_{t_{\max}}$
  for $t = 1, \ldots, t_{\max}$ do
      Sample mini-batch $S_t \leftarrow \mathsf{Pois}_p(S)$

      $\mathsf{grad}_t \leftarrow \nabla_\theta \ell(z;\theta_{t-1})$
      $\tilde{\mathsf{grad}}_t \leftarrow \frac{1}{|S_t|}\sum_{z\in S_t} \underbrace{{}^{1}\!/\max\{1,\, C^{-1}\cdot\|\mathsf{grad}_t\|_2\}}_{\text{Gradient clipping}} \cdot \mathsf{grad}_t + \underbrace{\mathcal{N}(0, \sigma^2\, C^2\, I_d)}_{\text{Gradient noise}}$
      $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \tilde{\mathsf{grad}}_t$
```

algorithm, the maximum discrepancy in terms of the L_2 distance between parameter vectors obtained by training on any two neighbouring datasets S and S'.

Output perturbation is not commonly used in practice on its own, as adding noise directly to the parameter vector can significantly deteriorate the error of the predictive model [33]. It is, however, used as a building block for more complicated methods to achieve DP.

DP-SGD. DP-SGD [1] is a standard method to achieve DP when training complex ML models such as neural networks.

DP-SGD, given in Algorithm 2, is a modification of SGD that achieves privacy by controlling the amount of information that is transferred from the training data to the parameter vector at each step. First, in order to ease the analysis of the privacy properties, the algorithm samples a mini-batch from the dataset using *Poisson sampling*, that is, every example $z \in S$ has the same probability $p \in [0, 1]$ of being sampled into the batch. We denote this sampling procedure as $\mathsf{Pois}_p(S)$. Second, each gradient vector is clipped to have a maximum L_2 norm of at most a given parameter C. Third, the algorithm adds noise sampled from an isotropic Gaussian distribution to the clipped gradient. One can view this as an application of the output perturbation mechanism above to each step of SGD.

1.4 Adversarial Robustness

Predictive models can operate in adversarial settings where adversaries might want to manipulate their feature vectors x in order to obtain a desired prediction. For instance, $\mathbb{X}$ could be a space of images, and $f_\theta(\cdot)$ a classifier of banned image content used by a social-media website to screen uploaded images. Then, there could exist an entity—the adversary—that wants to modify x containing banned content into x^* in a way that does not interfere with how the image is perceived by humans, yet does flip the classifier's prediction $f_\theta(x^*)$ in order to evade detection. In image domains, assuming $\mathbb{X} = \mathbb{R}^d$, given a parameter vector θ and an *initial example* $(x, y) \in \mathbb{X} \times \mathbb{Y}$, such task is commonly formalized using the following optimization problem to find an *adversarial perturbation* [123]:

$$\delta^* \in \arg\max_{\delta \in \mathbb{R}^d} \ell((x + \delta, y), \theta) \quad \text{s.t.} \quad \|\delta\|_p \leq \varepsilon. \tag{1.27}$$

The *adversarial example* x^* is obtained by applying the perturbation: $x^* = x + \delta^*$. The parameters p and ε define in which way the adversary wants the example x^* to be similar to the initial x. Informally, the adversary aims to maximize the loss incurred by x^* while keeping x^* within a certain distance from the initial example x. The crucial assumption in this definition is that any small perturbation within ε distance from an initial example (x, y) does not distort the example enough to change its semantics and the true label y.

Let us denote the *attack*, i.e., an algorithm which outputs an adversarial example for a given initial example z as $\mathcal{A}_\theta(z)$, omitting the parameters p and ε for conciseness. Then, we can measure robustness of a model to adversarial examples with *adversarial (or robust)* error:

$$R_{\mathcal{A}}(\theta) \triangleq \mathop{\mathbb{E}}_{z \sim P}[\ell(\mathcal{A}_\theta(z); \theta)]. \tag{1.28}$$

In the case that $\mathcal{A}_\theta((x, y)) = x + \delta^*$ solves Eq. (1.27) exactly, the adversarial error has the following closed form:

$$R_{\mathcal{A}}(\theta) = \mathop{\mathbb{E}}_{z \sim P}[\max_{\|\delta\|_p \leq \varepsilon} \ell((x + \delta, y), \theta)]. \tag{1.29}$$

Projected gradient descent. One way to solve the optimization problem in Eq. (1.27) is using *projected gradient descent* (PGD) [123], described in Algorithm 3. The algorithm repeatedly makes steps in the direction of the gradient of the loss, followed by a projection that ensures that the perturbation stays within the L_p constraint. Note that the gradient is computed with respect to the perturbation and not the parameters θ as is done in SGD. The projection operator finds the closest vector to a given input that satisfies the L_p constraint in Eq. (1.27):

$$\mathsf{Proj}_{x,p,\varepsilon}(x') \in \arg\min_{\bar{x} \in \mathbb{R}^d} \|\bar{x} - x'\|_2 \quad \text{s.t.} \quad \|\bar{x} - x\|_p \leq \varepsilon. \tag{1.30}$$

Adversarial training. A common way to mitigate the risks posed by adversarial examples is *adversarial training* [71], which means training on adversarial examples using the initial labels. This can be formalized [123] as finding model parameters that

Algorithm 3 PGD Attack

Input: Initial example (x, y), step size α, bound ε, number of iterations $t_{\max}$
Output: Adversarial example $x_{t_{\max}}$

```
$x_t \leftarrow x$
for $t$ in $1, \dots, t_{\max}$ do
    $\mathsf{grad}_t \leftarrow \nabla_\delta \ell((x_{t-1} + \delta, y); \theta)$
    $x_t \leftarrow \mathsf{Proj}_{x,p,\varepsilon}(x_{t-1} + \alpha \cdot \mathsf{grad}_t)$
```

Algorithm 4 Adversarial Training

Input: Dataset S, loss function $\ell(z; \theta)$, initial parameters θ_0, learning rate η, number of steps $t_{\max}$, mini-batch size b, attack algorithm $\mathcal{A}_\theta(\cdot)$
Output: Parameters of the trained model $\theta_{t_{\max}}$

```
for $t = 1, \dots, t_{\max}$ do
    Sample mini-batch $S_t \leftarrow \mathsf{Sample}_b(S)$
    $S_t^* \leftarrow \{\}$
    for $z \in S_t$ do
        $S_t^* \leftarrow S_t^* \cup \{\mathcal{A}_\theta(z)\}$
    $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \frac{1}{b} \sum_{z^* \in S_t^*} \nabla_\theta \ell(z^*; \theta_{t-1})$
```

minimize the empirical version of adversarial error:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{(x,y) \in S} \ell(\mathcal{A}_\theta(z); \theta) \tag{1.31}$$

A way to solve this optimization problem is using a modification of SGD described in Algorithm 4. Instead of computing the loss gradients with respect to an example $z \in S$ as in standard SGD, in adversarial training the gradients are computed with respect to adversarial examples $\mathcal{A}_\theta(z)$ obtained, e.g., by running the PGD attack in Algorithm 3.

Chapter 2

Privacy and Reliable Learning

This chapter is based on a peer-reviewed article entitled “What You See is What You Get: Principled Deep Learning via Distributional Generalization” [107] by Bogdan Kulynych, Yao-Yuan Yang, Yaodong Yu, Jarosław Błasiok, and Preetum Nakkiran, published in 2022 Advances in Neural Information Processing Systems (NeurIPS).

The concept of distributional generalization (DG) was introduced concurrently by Nakkiran and Bansal [129] and Kulynych et al. [106]. The latter work is the basis of Chapter 4. Although DG was not initially introduced in the paper on which the present chapter is based, we first describe it in Section 2.2 as opposed to Chapter 4 for clarity.

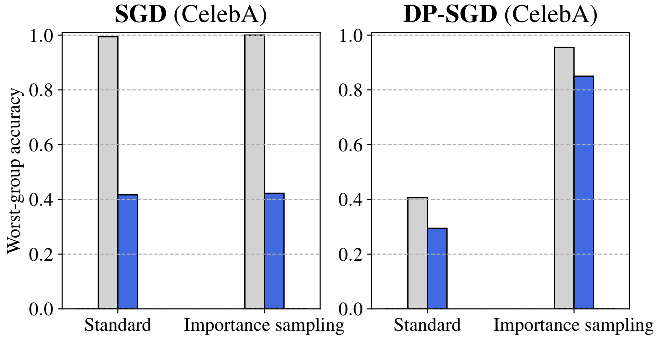


Figure 2.1: **Differential privacy ensures the desired behavior of importance sampling on test data.** The train and test accuracy of ResNets on CelebA, evaluated on the worst-performing ("male, blond") subgroup. *Left:* Standard SGD has a large generalization gap on this subgroup, and Importance Sampling (IS) has little effect. *Right:* DP-SGD provably has small generalization gap on all subgroups, and IS improves subgroup performance as intended. See Section 2.5 for details.

2.1 Introduction

Much of machine learning (ML), both in theory and in practice, operates under two assumptions. First, we have independent and identically distributed (i.i.d.) samples. Second, we care only about a single averaged scalar metric (error, loss, risk). Under these assumptions, we have mature methods and theory: Modern learning methods excel when trained on i.i.d. data to directly optimize a scalar loss, and there are many theoretical tools for reasoning about *generalization*, which explain when does optimization of a scalar on the training data translates to similar values of this scalar at test time.

The focus on scalar metrics such as average error, however, misses many theoretically, practically, and socially relevant aspects of model performance. For example, models with small *average* error often have high error on salient minority subgroups [24, 98]. In general, ML models are applied to the heterogeneous and long-tailed data distributions of the real world [171]. Attempting to summarize their complex behavior with only a single scalar misses many rich and important aspects of learning.

These issues are compounded for modern overparameterized networks, as their nuanced test-time behavior is not reflected at training time. For example, consider the setting of *importance sampling*: suppose we know that a certain subgroup of inputs is underrepresented in the training data compared to the test distribution (breaking the i.i.d. assumption). For underparameterized models, we can upsample this underrepresented group to account for the distribution shift [see, e.g., 74]. This approach, however, is known to empirically fail for overparameterized models [25]. Because "what you see" (on the training data) is not "what you get" (at test time), we cannot make principled train-time interventions to affect test-time behaviors. This issue extends beyond importance sampling. For example, theoretically principled methods for distributionally robust optimization (e.g. Namkoong and Duchi [130]) fail for overparameterized deep networks, and require ad-hoc modifications [151].

We develop a theoretical framework which sheds light on these existing issues, and leads to improved practical methods in privacy, fairness, and distributional robustness. The core object in our framework is what we call the "What You See Is What You Get" (WYSIWYG) property. A training procedure with the WYSIWYG property does *not* exhibit the "pathologies" of standard stochastic gradient descent (SGD): all test-time behaviors will be expressed on the training data as well, and there will be "no surprises" in generalization.

What You See Is What You Get (WYSIWYG) as a Design Principle. The WYSIWYG property is desirable for two reasons. The first is diagnostic: as there are "no surprises" at test time, all properties of a model at test time are already evident at the training stage. It cannot be the case, for example, that a WYSIWYG model has small disparate impact on the training data, but large disparate impact at test time. The second reason is algorithmic: to mitigate *any* unwanted test-time behavior, it is sufficient to mitigate this behavior on the training data. This means that algorithm designers can be concerned only with achieving desirable behavior at train time, as the WYSIWYG property guarantees it holds at test time too. In practice, this enables the usage of many theoretically principled algorithms which were developed in the underparameterized regime to also apply in the modern overparameterized (deep learning) setting. For example, we find that interventions such as importance sampling, or algorithms for distributionally robust optimization, which fail without additional regularization, work exactly as intended with WYSIWYG (See Fig. 2.1 for an illustration).

As WYSIWYG is a high-level conceptual property, we have to formalize it to use in computational practice. We do so using the notion of *Distributional Generalization* (DG). If classical generalization ensures that the values of the model's loss on the training dataset and at test time are close on average [159], distributional generalization ensures that values of any other bounded test function—not only loss—are close on training and test time. We say that a model which satisfies an appropriately high level of distributional generalization exhibits the WYSIWYG property.

Achieving DG in Practice. Our key observation is that distributional generalization is formally implied by differentially private (DP) training (see Section 1.3). The spirit of this observation is not novel: DP training is known to satisfy much stronger notions of generalization (e.g., *robust generalization*, see Section 2.6 for more details), and stability than standard SGD [13, 44, 57, 164]. We show that a similar connection holds for the notion of distributional generalization, and prove (and improve) tight bounds relating DP, stability, and DG. This guarantees the WYSIWYG property for any method that is differentially-private, including DP-SGD on deep neural networks [1].

We demonstrate how DG can be a useful design principle in three concrete settings. First, we show that we can mitigate disparate impact of DP training [8, 145] by leveraging importance sampling. Second, we study the setting of distributionally robust optimization [e.g., 86, 151]. We show how ideas from DP can be used to construct heuristic optimizers, which do not formally satisfy DP, yet empirically exhibit DG. Our heuristics lead to competitive results with SOTA algorithms in five datasets in the distributional robustness setting. Third, we show that the heuristic optimizer is also capable of reducing overfitting of adversarial loss in adversarial training [123, 147, 197].

Our Contributions. We develop the theoretical connection between Differential Privacy (DP) and Distributional Generalization (DG), and we leverage our theory to improve empirical performance in privacy, fairness, and robustness applications. Theoretically (Sections 2.2 to 2.4):

1. We provide tighter bounds than previously reported connecting DP and strong forms of generalization, and show that DP training methods satisfy DG, thus the WYSIWYG property.
2. We introduce DP-IS-SGD, an importance-sampling version of DP-SGD, and show it satisfies DP and DG.

Experimentally (Section 2.5):

1. We use our framework to shed light on *disparate impact*: The disparity in accuracy across groups at test time is provably reflected by the accuracy disparity *on the train dataset.*
2. We use our DP-IS-SGD algorithm to largely mitigate the disparate impact of DP using importance sampling.
3. Based on our theoretical intuitions, we propose a DP-inspired heuristic: addition of gradient noise. We find this empirically achieves competitive and even improved results in several DRO settings, and reduces overfitting of adversarial loss in adversarial training.

Taken together, our results emphasize the central role of the WYSIWYG property in designing machine learning algorithms which avoid the "pathologies" of standard SGD. We also establish DP as a useful tool for achieving WYSIWYG, thus extend its applications further beyond privacy.

2.2 Theory of "What You See is What You Get" Generalization

We first review the notion of distributional generalization and demonstrate why it captures the WYSIWYG property. Second, we show that strong stability notions imply distributional generalization. Finally, we improve on the known stability guarantees of differential privacy. As a result, we extend the connections between differential privacy, stability, and generalization to *distributional* generalization, showing that stability and privacy imply the WYSIWYG property.

2.2.1 Distributional Generalization and WYSIWYG

If on-average generalization (see Section 1.2) guarantees closeness only of loss values on train and test data, distributional generalization (DG) also guarantees closeness of values of all test functions $\phi(z;\theta) \in [0,1]$ beyond only loss:

Definition 2.1 (Based on Nakkiran and Bansal [129])**.** An algorithm $T(S)$ satisfies δ-distributional generalization if for all $\phi : \mathbb{D} \times \Theta \to [0,1]$,

$$\left| \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim S}} \phi\Big(z; T(S)\Big) - \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim P}} \phi\Big(z; T(S)\Big) \right| \leq \delta. \tag{2.1}$$

By the variational form of the TV distance (see Section 1.1), Eq. (2.1) is equivalent to the bound $d_{\mathsf{TV}}(P_1, P_0) \leq \delta$, where P_1 and P_0 are both distributions of $\Big(z, T(S)\Big)$ over the randomness of $S \sim P^n$ and the training algorithm $T(\cdot)$, with the difference that $z \sim S$ in the case of P_1 (train), and $z \sim P$ in the case of P_0 (test).

It might seem that DG only ensures average closeness of bounded tests on train and test data. This is not, however, the full picture. Consider generalization in terms of a broader class of functions:

Definition 2.2. An algorithm $T(S)$ satisfies (δ, π)-distributional generalization if for a given property function $\pi : \mathbb{D} \times \Theta \to \mathbb{R}^k$ it holds that $d_{\mathsf{TV}}(\pi_\sharp P_1, \pi_\sharp P_0) \leq \delta$.

Because bounds on TV distance are preserved under post-processing, we can see that δ-distributional generalization implies (δ, π)-distributional generalization for *all* property functions. Informally, δ-DG means that for *all* numeric property functions $\pi(z;\theta)$ of a model, the distributions of the property values are close on the train and test data, on average. This fact captures the high-level idea of the *"What You See is What You Get"* (WYSIWYG) guarantee. Some example property functions:

- *Subgroup loss:* $\pi(z;\theta) = \mathbb{1}[z \in G] \cdot \ell(z;\theta)$, for some subgroup $G \in \mathbb{G}$.
- *Counterfactual fairness:* $\pi((x, y);\theta) = f_\theta(x') - f_\theta(x)$, where x' is a counterfactual version of x had it had a different value of a sensitive attribute [109].
- *Robustness to corruptions:* $\pi(z;\theta) = \ell(\mathcal{A}(z);\theta)$, where $\mathcal{A}(x)$ is a possibly randomized transformation that distorts the example, e.g., by adding Gaussian noise.
- *Adversarial robustness:* $\pi(z;\theta) = \ell(\mathcal{A}_\theta(z);\theta)$, where $\mathcal{A}_\theta(z)$ is an adversarial example, e.g. generated using the PGD attack (see Section 1.4).

In the next sections, we show how a training algorithm can provably satisfy DG and therefore provide WYSIWYG guarantees for all properties, including the ones above.

2.2.2 Distributional Generalization from Stability and Differential Privacy

The connections between privacy, stability, and generalization are well-known. In particular, stability of the learning algorithm—its non-sensitivity to limited changes in the training data—implies generalization [20, 159]. In turn, differential privacy implies strong forms of stability, thus ensuring generalization through the chain Privacy ⇒ Stability ⇒ Generalization [57, 58, 146, 178].

DP mathematically encodes a notion of plausible deniability of the inclusion of an example in the dataset. However, it can also be thought as a strong form of stability [58]. As such, DP implies other notions of stability. We consider the following notion, which has been studied in the literature under multiple names. In the context of privacy, it is equivalent to $(0, \delta)$-differential privacy, and has been called additive differential privacy [67], and total-variation privacy [10]. In the context of learning, it has been called total-variation (TV) stability [13]. We take this last approach and refer to it as TV stability:

Definition 2.3 (TV Stability)**.** An algorithm $T(S)$ is δ-TV stable if for any two *neighbouring datasets* S, S' of size n, for any subset $V \subseteq \Theta$ it holds that

$$\Pr[T(S) \in V] \leq \Pr[T(S') \in V] + \delta. \tag{2.2}$$

Equivalently, $d_{\mathsf{TV}}(T(S), T(S')) \leq \delta$.

It is easy to see that (ϵ, δ)-DP immediately implies δ'-TV stability with:

$$\delta' = \exp(\epsilon) - 1 + \delta. \tag{2.3}$$

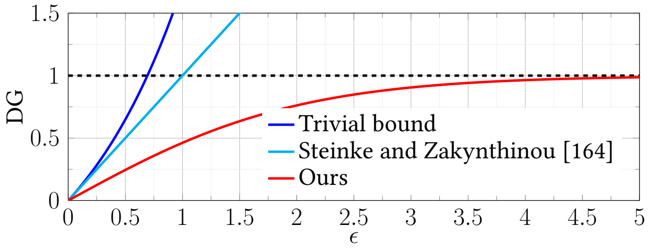


Figure 2.2: DG bound from ϵ-DP.

From Classical to Distributional Generalization. Similarly to the classical generalization, one way to achieve distributional generalization is through strong stability:

Theorem 2.1. Suppose that the training algorithm is δ-TV stable. Then, the algorithm satisfies δ-DG.

We refer to Appendix A.1 for the proofs of this and all other formal statements in the rest of the chapter.

As DP implies TV-stability, by Theorem 2.1 we have that DP also implies DG. We show that DP algorithms enjoy a significantly stronger stability guarantee than previously known, which means that the DG guarantee that one obtains from DP is also stronger.

Proposition 2.2.1. An algorithm which is (ϵ, δ)-DP is also δ'-TV stable with:

$$\delta' = \frac{\exp(\epsilon) - 1 + 2\delta}{\exp(\epsilon) + 1}.$$

In Appendix B.1.2, we discuss the relationship of this result to other works in the literature on information-theoretic generalization. In particular, to Steinke and Zakynthinou [164] whose results can also be used to relate DP and DG. Fig. 2.2 shows that the known bounds quickly become vacuous unlike the bound in Proposition 2.2.1. In fact, we show that our bound is tight in Appendix A.1.

Stronger Distributional Generalization Guarantees. Although DG immediately implies generalization for all bounded properties, it is possible to obtain tighter bounds from TV stability. For example, directly applying DG to the *subgroup loss* property yields a bound that decays with the size of the subgroup: accuracy on very small subgroups is not guaranteed to generalize well.

2.3 Example Applications

To demonstrate that WYSIWYG is a useful property in algorithm design, in the remainder of this chapter we use it to construct simple and high-performing algorithms for three example applications: mitigation of disparate impact of DP, ensuring group-distributional robustness, and mitigation of robust overfitting in adversarial training.

Mitigating Disparate Impact of DP. First, we consider applications in which learning presents privacy concerns, e.g., in the case that the training data contains sensitive information. Using training procedures that satisfy DP is a standard way to guarantee privacy in such settings. Training with DP, however, is known to incur *disparate impact* on the model accuracy: some subgroups of inputs can have worse test accuracy than others. For example, Bagdasaryan et al. [8] show that using DP-SGD—a standard algorithm for satisfying DP [1]—in place of regular SGD causes a significant accuracy drop on “darker skin” faces in models trained on the CelebA dataset of celebrity faces [117], but a less severe drop on “lighter skin” faces. Our goal is to mitigate such disparate impact. This issue—a quality-of-service harm [122]—is but one of many possible harms due to ML systems. We do not aim to mitigate any other broad fairness-related issues, nor claim this is possible within our framework.

For given parameters (ϵ, δ), we want to learn a model θ that simultaneously satisfies (ϵ, δ)-DP, has high overall accuracy, and incurs small *loss disparity*:

$$\max_{G,G' \in \mathbb{G}} \left| \mathop{\mathbb{E}}_{z \sim P_G} [\ell(z;\theta)] - \mathop{\mathbb{E}}_{z \sim P_{G'}} [\ell(z;\theta)] \right| . \tag{2.4}$$

Group-Distributional Robustness. Next, we consider a setting of *group-distributionally robust optimization* [e.g., 86, 151]. If in the standard learning approach we want to train a model that minimizes *average* loss, in this setting, we want to minimize the *worst-case (highest) group loss.* This objective can be used to mitigate fairness concerns such as those discussed previously, as well as to avoid learning spurious correlations [151].

Formally, we want to learn a model θ that minimizes the *worst-case group loss*:

$$\max_{G \in \mathbb{G}} \mathop{\mathbb{E}}_{z \sim P_G} [\ell(z;\theta)] . \tag{2.5}$$

Unlike the previous application, in this setting, we do not require privacy of the training data. We use training with DP as a *tool* to ensure the generalization of the worst-case group loss.

Mitigating Robust Overfitting. Finally, we consider the setting of robustness to test-time adversarial examples through adversarial training (see Section 1.4):

$$\mathop{\mathbb{E}}_{z \sim P}[\ell(\mathcal{A}_\theta(z); \theta)]. \tag{2.6}$$

Rice et al. [147] observed that adversarially trained models exhibit "robust overfitting": higher generalization gap of robust loss than that of the regular loss. In this application, we similarly aim to use a relaxed version of training with DP as a tool to ensure generalization of robust loss, thus mitigate robust overfitting.

2.4 Algorithms which Distributionally Generalize

In this section, we construct algorithms for the applications in Section 2.3. Our approach follows the blueprint: First, we apply a principled algorithmic intervention that ensures desired behavior on *the training data* (e.g., importance sampling). Second, we modify the resulting algorithm to additionally ensure DG, which guarantees that the desired behavior generalizes to *test time*.

2.4.1 DP Training with Importance Sampling

Our first algorithm, DP-IS-SGD (Algorithm 5), is a version of DP-SGD [1] which performs importance sampling. DP-IS-SGD is designed to mitigate disparate impact while retaining DP guarantees. The standard DP-SGD samples data batches using *uniform Poisson subsampling:* Each example in the training set is chosen into the batch according to the outcome of a Bernoulli trial with probability $\bar{p} \in [0, 1]$. To correct for unequal representation and the resulting disparate impact, we use *non-uniform Poisson subsampling*: Each example $z \in S$ has a possibly different probability $p(z)$ of being selected into the batch, where $p(z)$ does not depend on the dataset S otherwise, and is bounded: $0 \leq p(z) \leq p^* \leq 1$. We denote this subsampling procedure as $\mathsf{Pois}_{p(\cdot)}(S)$.

We choose $p(z)$ to satisfy two properties. First, to increase the sampling probability for examples in minority groups: $p(z) \propto {}^1\!/_{q_{g(z)}}$. Second, to keep the average batch size equal to $\bar{p} \cdot n$ as in standard DP-SGD. In the rest of the chapter, we assume that the group probabilities $(q_1, \ldots, q_m)$ are known, but it is possible to estimate them in a private way using standard methods [133]. We present DP-IS-SGD in Algorithm 5, along with its differences to the standard DP-SGD.

DP Properties of DP-IS-SGD. Uniform Poisson subsampling is well-known to amplify the privacy guarantees of an algorithm [32, 114]. For example, Li et al. [114] show that if an algorithm $T(S)$ satisfies (ϵ, δ)-DP, then $T \circ \mathsf{Pois}_{\bar{p}}(S)$ provides approximately $(O(\bar{p}\epsilon), \bar{p}\delta)$-DP for small values of ϵ. We show in Appendix A.1 that non-uniform Poisson subsampling provides the same amplification guarantee with $\bar{p} = p^*$, where p^* is the maximum value of $p(\cdot)$.

Algorithm 5 DP-IS-SGD (DP Importance Sampling SGD)

```
Input: Dataset $S$, loss $\ell(z;\theta)$, initial parameters $\theta_0$, learning rate $\eta$, maximal
  gradient norm $C$, noise parameter $\sigma$, number of steps $t_{\max}$, sampling rate $\bar{p}$,
  group probabilities $(q_1, \ldots, q_m)$ .
  for $t = 1, \ldots, t_{\max}$ do
    Sample batch $S_t \leftarrow$ $\mathsf{Pois}_{p(\cdot)}(S)$, with sampling probabilities $p(z) \triangleq \bar{p}/m \cdot q_{g(z)}$
    $\tilde{\mathsf{grad}}_t \leftarrow \frac{1}{|S_t|} \sum_{z \in S_t} \underbrace{1/\max\{1,\, C^{-1} \cdot \|\nabla_\theta \ell(z;\theta_{t-1})\|_2\}}_{\text{Gradient clipping}} \cdot \nabla_\theta \ell(z;\theta_{t-1}) + \underbrace{\mathcal{N}(0, \sigma^2\, C^2\, I_d)}_{\text{Gradient noise}}$
    $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \tilde{\mathsf{grad}}_t$
```

The highlighted parts indicate the differences with respect to DP-SGD. We obtain DP-SGD as a special case when we have a single group with $q = 1$ (implying $p(z) = \bar{p}$).

As this guarantee is independent of the internal workings of $T(S)$, it is loose. For DP-SGD, one way of computing tight privacy guarantees of subsampling is using the notion of *Gaussian differential privacy* (GDP) [51]. GDP is parameterized by a single parameter μ. If an algorithm $T(S)$ satisfies μ-GDP, one can efficiently compute a set of (ϵ, δ)-DP guarantees also satisfied by $T(S)$ [51]. We show that we can use any GDP-based mechanism for computing the privacy guarantee of DP-SGD to obtain the privacy guarantees of DP-IS-SGD in a black-box manner:

Proposition 2.4.1. Let us denote by $\mu(\bar{p}, \sigma, C, T)$ (see Algorithm 5) a function that returns a μ-GDP guarantee of DP-SGD. Then, DP-IS-SGD satisfies a GDP guarantee $\mu(p^*, \sigma, C, T)$.

2.4.2 Gaussian Gradient Noise

We showed that DP-IS-SGD enjoys theoretical guarantees for both DP and DG. DP models, however, often have lower test accuracy compared to standard training [33]. This can be an unnecessary disadvantage in settings where privacy is not required, such as in our robustness applications. Thus, we explore training algorithms which are inspired by our theory yet do not come with generic theoretical guarantees of DG.

Note that DP-SGD uses gradient *clipping* and *noise* (see Algorithm 5). Individually, these are used as *regularization* for improving stability and generalization [79, 132]. Following this, we relax DP-IS-SGD to only use gradient noise. This sacrifices privacy guarantees in exchange for practical performance. Specifically, we apply gradient noise to three standard algorithms for achieving group-distributional robustness: importance sampling (IS-SGD), importance weighting (IW-SGD) [74], and gDRO [151]. This results in the following variations: IS-SGD-n, IW-SGD-n, gDRO-n, respectively. Similarly, we apply gradient noise to standard PGD adversarial training [123]. See Appendix B.2 for additional details.

2.5 Experiments

We empirically study the distributional generalization in real-world applications.

Datasets. We use the following datasets with group annotations: CelebA [117], UTKFace [202], iNaturalist2017 (iNat) [84], CivilComments [19], MultiNLI [151, 184], and ADULT [100]. For every dataset, each example belongs to one group (e.g., CelebA) or multiple groups (e.g., CivilComments). For example, in the CelebA dataset, there are four groups: "blond male", "male with other hair color", "blond female", and "female with other hair color". Additionally, we use the CIFAR-10 [103] dataset for the adversarial-overfitting application. We present more details on the datasets, their groups, and used model architectures in Appendix B.3.

2.5.1 Enforcing DG in Practice

We empirically confirm that a training procedure with DP guarantees also has a bounded DG gap.

In practice, it is not possible to compute the exact DG gap. As a proxy in applications which concern subgroup performance in this section, and Sections 2.5.2 and 2.5.3, we use the difference between train-time and test-time worst-group accuracy. This (a) follows the empirical approach by Nakkiran and Bansal [129] which proposes to estimate the gap in Eq. (2.1) using a finite set of test functions, and (b) measures the aspect of distributional generalization that is relevant to our applications. We provide more details on this choice of the proxy measure in Appendix B.3.2.

We train a model on CelebA using DP-SGD for varying levels of ϵ. Fig. 2.3 shows that the gap between training and testing worst-group accuracy increases as the level of privacy decreases, which is consistent with our theoretical bounds.

2.5.2 Disparate Impact of Differentially Private Models

We evaluate DP-IS-SGD (Algorithm 5), and demonstrate that it can mitigate the disparate impact in realistic settings where both privacy and fairness are required.

Fig. 2.4 shows the accuracy disparity, test accuracy, and worst-case group accuracy, computed as in Eq. (2.4), as a function of the privacy budget ϵ. The models are trained with DP-SGD and DP-IS-SGD. When comparing DP-SGD and DP-IS-SGD with the same or similar ϵ, we observe that DP-IS-SGD achieves lower disparity on all datasets. However, this comes with a drop in average accuracy. On CelebA, with $\epsilon \in [2, 12]$,

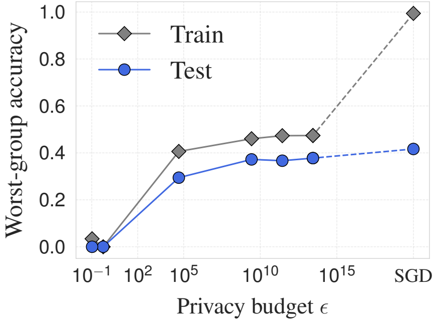
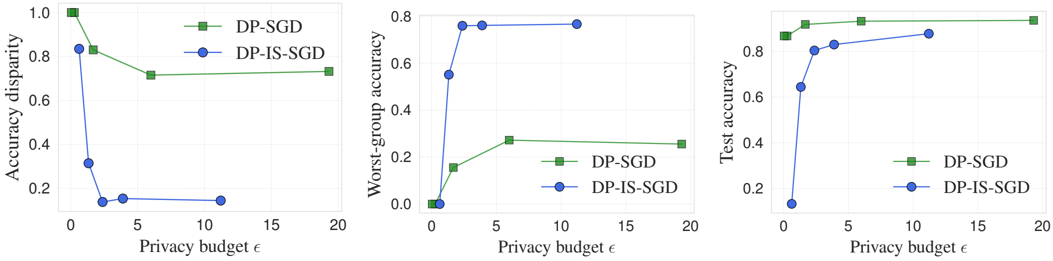


Figure 2.3: **Privacy induces DG.** Train/test worst-case group accuracies as a function of privacy parameter ϵ of DP-SGD on CelebA (x axis). Increasing privacy reduces the generalization gap.



(a) Accuracy disparity (lower is better)

(b) Worst-group accuracy (higher is better)

(c) Test accuracy (higher is better)

Figure 2.4: **Importance Sampling Improves Disparate Impact of DP-SGD.** The accuracy disparity of the models trained with DP-SGD and DP-IS-SGD on CelebA. Adding importance sampling (IS) improves disparate impact at most privacy budgets in this setting. We set $\delta = 1/2n$, where n is the number of training examples. We use GDP accountant to compute the privacy budget ϵ.

DP-IS-SGD has around 8 p.p. lower test accuracy than DP-SGD. At the same time, the disparity drop ranges from 40 p.p. to 60 p.p., which is significantly higher than the accuracy drop. We observe similar results on UTKFace. On iNat, however, although DP-IS-SGD decreases disparity, the overall test accuracy suffers a significant hit. This is likely because the minority subgroup is very small, which results in high maximum sampling probability p^*, thus deteriorating the privacy guarantee. Details for UTKFace and iNat are in Appendix B.3.3.

In summary, we find that DP-IS-SGD can achieve lower disparity at the same privacy budget compared to standard DP-SGD, with mild impact on test accuracy.

Comparison to DP-SGD-F [192]. DP-SGD-F is a variant of DP-SGD which dynamically adapts gradient-clipping bounds for different groups to reduce the disparate impact. We did not manage to achieve good overall performance of DP-SGD-F on the datasets above. In Appendix B.3.3, we compare it to DP-IS-SGD on the ADULT dataset (used by Xu et al. [192]), finding that DP-IS-SGD obtains lower disparity for the same privacy level, yet also lower overall accuracy.

2.5.3 Group-Distributionally Robust Optimization

We investigate whether our proposed versions of standard algorithms with Gaussian gradient noise (Section 2.4.2) can improve group-distributional robustness. To do so, we evaluate empirical DG using worst-group accuracy as a proxy for DG gap as in Section 2.5.1, following the evaluation criteria in prior work [88, 151]. State-of-the-art (SOTA) methods apply L_2 regularization and early-stopping to achieve the best performance. We compare three baselines with L_2 regularization, IS-SGD-L_2, IW-SGD-L_2, and gDRO-L_2 to our noisy-gradient variations as well as DP-IS-SGD. We use the validation set to select the best-performing regularization parameter and epoch (for early stopping) for each method. See Appendix B.3.4 for details on the experimental setup.

Table 2.1 shows the worst-group accuracy of each algorithm on five datasets. When comparing IS-SGD, IW-SGD, and gDRO with their noisy counterparts, we observe that the noisy versions in general have similar or slightly better performance compared to non-noisy counterparts. For instance, IS-SGD-n improves the SOTA results on CivilComments dataset. This showcases that in terms of learning distributionally robust models, *noisy gradient can be a more effective regularizer than the currently standard L_2 regularizer.* We also find that DP-IS-SGD improves on baseline methods or even achieves SOTA-competetitive performance on several datasets. For instance, on CelebA and MNLI, DP-IS-SGD achieves better performance than IS-SGD-ℓ_2. This is surprising, as DP tends to deteriorate performance. This suggests that distributional robustness and privacy are not incompatible goals. Moreover, DP can be a useful tool even when privacy is not required.

2.5.4 Mitigating Robust Overfitting

As in the previous section, we expect that a modification of a standard projected gradient descent (PGD) method for adversarial training [123] with added Gaussian gradient noise (Section 2.4.2) improves the generalization behavior of adversarial training.

To verify this, we adversarially train models on the CIFAR-10 dataset with varying levels of the noise magnitude. We provide more details on the setup in Appendix B.3.5. Fig. 2.5 shows that in standard adversarial training without noise the gap between robust training accuracy and robust test accuracy is large at approximately 30 p.p., which is consistent with the prior observations of Rice et al. [147]. By injecting noise

Table 2.1: **Our noisy-gradient algorithms produce competitive results compared to counterparts with L_2 regularization.** The table shows the worst-group accuracy of each algorithm. Baselines are in the top rows; our algorithms are in the bottom. For gDRO-L_2-SOTA, we show avg. ± std. over five runs from Idrissi et al. [88]. For CelebA, we show avg. ± std. over three random splits.

	CelebA	UTKFace	iNat.	Civil.	MNLI
SGD-L_2	73.0 ± 2.2	86.3	41.8	57.4	67.9
IS-SGD-L_2	82.4 ± 0.5	85.8	70.6	64.3	70.4
IW-SGD-L_2	**89.0** ± 0.9	86.5	67.6	65.7	68.1
gDRO-L_2	84.5 ± 0.8	85.2	67.3	67.3	75.9
gDRO-L_2-SOTA	86.9 ± 0.5	–	–	69.9 ± 0.5	**78.0** ± 0.3
DP-IS-SGD	86.0 ± 0.8	82.5	51.4	70.4	72.3
IS-SGD-n	84.9 ± 1.0	85.5	**71.0**	**71.9**	70.8
IW-SGD-n	**88.5** ± 0.4	**88.5**	70.9	69.9	69.7
gDRO-n	83.3 ± 0.5	87.5	56.4	71.3	**78.0**

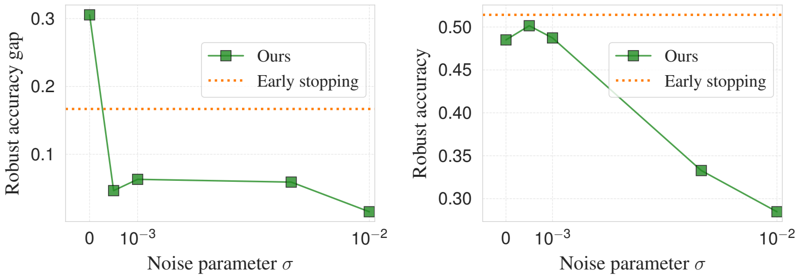
into the gradient, our proposed approach decreases the generalization gap of robust accuracy by more than 3× to less than 10 p.p. Surprisingly, in our experiments, training with gradient noise achieves both a small adversarial accuracy gap *and* better adversarial test accuracy compared to standard adversarial training, when using a small noise magnitude ($\sigma = 5 \times 10^{-4}$). In terms of resulting robust accuracy, the method's performance is comparable to early stopping, identified as the most effective way to prevent robust overfitting by Rice et al. [147]. These experimental results demonstrate how WYSIWYG can be a useful design principle in practice.

2.6 Related Work

DP and Strong Generalization. DP is known to imply a stronger than standard notion of generalization, called *robust generalization*[1] [13, 44]. Robust generalization can be thought as a high-probability counterpart of DG: generalization holds with high probability over the train dataset, not only on average over datasets. We focus on our notion of DG for both conceptual and theoretical simplicity. A more comprehensive discussion of relations to robust generalization is in Appendix B.1.1. Other than robust generalization, our connections in Section 2.2 can also be derived from weaker generalization bounds that rely on information-theoretic measures [164]. We detail this in Appendix B.1.2.

Disparate Impact of DP. Bagdasaryan et al. [8], Pujol et al. [145] have shown that ensuring DP in algorithmic systems can cause error disparity across population groups.

[1] Unrelated to "robust overfitting" in adversarial training.



(a) Gen. gap of robust accuracy (lower is better)

(b) Robust accuracy (higher is better)

Figure 2.5: **Noisy gradient reduces overfitting in adversarial training.** We show the generalization gap of robust accuracy (left), and test-time robust accuracy (right) of adversarially trained models with different levels of noise magnitude. The dash orange lines represent the performance of adversarial training with early stopping. The model trained without noise exhibits “robust overfitting” of about 30 p.p. Gradient noise reduces the generalization gap by more than 3× for all values of the noise parameter at a cost of decreased robust accuracy as the noise gets larger.

Xu et al. [192] proposed a variant of DP-SGD for reducing disparate impact. We compare our method to DP-SGD-F in Appendix B.3.3. In another line of related work, Cummings et al. [45], Sanyal et al. [154] show fundamental trade-offs between model’s loss and DP training. As our theoretical results concern generalization, not loss per se, our results do not contradict these theoretical trade-offs. We discuss the relationship in detail in Appendix B.1.3.

Group-Distributional Robustness. Group-distributional robustness aims to improve the worst-case group performance. Existing approaches include using worst-case group loss [127, 151, 199], balancing majority and minority groups by reweighting or subsampling [25, 88, 152], leveraging generative models [70], and applying various regularization techniques [27, 151]. Although some work [27, 151] discusses the importance of regularization in distributional robustness, they have not explored potential reasons for this (e.g. via the connection to generalization). Another line of work studies how to improve group performance without group annotations [41, 53, 116], which is a different setting from ours as we assume the group annotations are known.

Robust Overfitting. Rice et al. [147], Yu et al. [196] have shown that adversarially trained models tend to overfit in terms of robust loss. Rice et al. [147] proposed to use regularization to mitigate overfitting, but the noisy gradient has not been explored for this. We showed that the WYSIWYG framework can serve as an alternative direction for mitigating and explaining this issue.

2.7 Conclusions and Future Work

We argue that a "What You See is What You Get" property, which we formalize through the notion of distributional generalization (DG), can be desirable for learning algorithms, as it enables principled algorithm design in settings including deep learning. We show that this property is possible to achieve with DP training. This enables us to leverage advances in DP to enforce DG in many applications.

We propose enforcing DG as a general design principle, and we use it to construct simple yet effective algorithms in three settings. In certain fairness settings, we largely mitigate the disparate impact of differential privacy by using importance sampling and enforcing DG in our new algorithm DP-IS-SGD. In our analysis, however, the privacy and DG guarantees of DP-IS-SGD deteriorate in the presence of very small groups. Future work could explore individual-level accounting [63] for a tighter analysis. In certain worst-case generalization settings, inspired by DP-SGD, we propose using a noisy-gradient regularizer. Compared to SOTA algorithms in DRO, noisy gradient achieves competitive results across many standard benchmarks. In certain adversarial-robustness settings, our proposed noisy-gradient regularizer significantly reduces robust overfitting. An interesting direction for future work would be to explore its effectiveness in large-scale settings, e.g., ImageNet [43]. We hope future work can explore extending this design principle to ensure generalization of other properties, such as calibration and counterfactual fairness.

Chapter 3

Privacy and Arbitrary Decisions

This chapter is based on a peer-reviewed article entitled “Arbitrary Decisions are a Hidden Cost of Differentially Private Training” [108] by Bogdan Kulynych, Hsiang Hsu, Carmela Troncoso, and Flavio du Pin Calmon, published in the proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT).

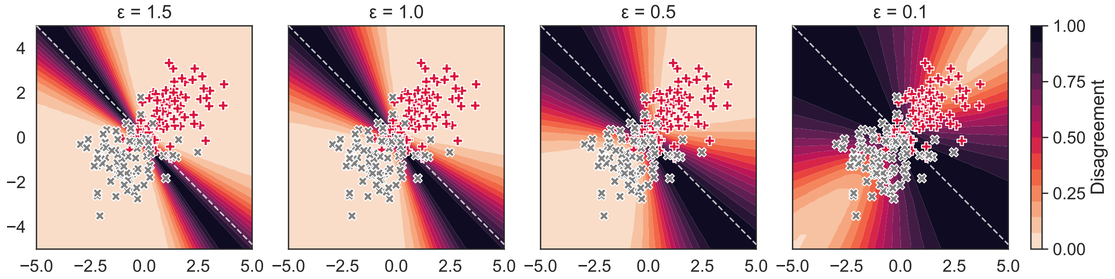


Figure 3.1: **The region of examples which exhibit high variance of decisions (dark) across similar models grows as the privacy level increases (lower ϵ).** Each plot shows the level of decision disagreement across $m = 5{,}000$ logistic-regression models (darker means higher disagreement) trained with varying levels of differential privacy (ϵ value, lower means more private) using the objective-perturbation method [33]. All models attain at least 72% accuracy on the test dataset (50% is the baseline). The disagreement value of 1.0 means that out of the m models, half output the positive decision, whereas the other half output the negative one for a given example. The values of disagreement are shown for different possible two-dimensional examples, with x and y axes corresponding to the two dimensions. The markers show training data examples belonging to two classes (denoted as $\times$ and $+$, respectively). Without DP, there is a single optimal classification model. The dotted line - - shows the decision boundary of this optimal non-private model. See Section 3.5 for details.

3.1 Introduction

In many high-stakes prediction tasks (e.g., lending, healthcare), training data used to fit parameters of machine-learning models are privacy-sensitive. As explained in Section 1.3, a standard technical approach to ensure privacy is to use training procedures that satisfy *differential privacy* (DP) [54, 56]. DP is a formal condition that, intuitively, guarantees a degree of plausible deniability on the inclusion of an individual sample in the training data. In order to satisfy this condition, non-trivial differentially-private training procedures use randomization (see, e.g., Abadi et al. [1], Chaudhuri et al. [33]). The noisy nature of DP mechanisms is key to guarantee plausible deniability of a record's inclusion in the training data. Unfortunately, randomization comes at a cost: it often leads to decreased accuracy compared to non-private training [89]. Reduced accuracy, however, is not the only cost incurred by differentially-private training. DP mechanisms can also increase *predictive multiplicity*, discussed next.

In a prediction task, there can exist multiple models that achieve comparable levels of accuracy yet output drastically different predictions for the same input. This phenomenon is known as predictive multiplicity [124], and has been documented in multiple realistic machine-learning settings [85, 124, 182]. Predictive multiplicity can appear due to under-specification and randomness in the model's training procedure [17].

Predictive multiplicity formalizes the *arbitrariness* of decisions based on a model's output. In practice, predictive multiplicity can lead to questions such as "*Why has a model issued a negative decision on an individual's loan application if other models with indistinguishable accuracy would have issued a positive decision?*" or "*Why has a model suggested a high dose of a medicine for an individual if other models with comparable average accuracy would have prescribed a lower dose?*" These examples highlight that acting on predictions of a single model without regard for predictive multiplicity can result in arbitrary decisions. Models produced by training algorithms that exhibit high predictive multiplicity face fundamental challenges to their credibility and justifiability in high-stakes settings [18, 59].

In this chapter, we demonstrate a fundamental connection between privacy and predictive multiplicity: For a fixed training dataset and model class, DP training results in models that ensure the same degree of privacy and achieve comparable accuracy, yet assign conflicting outputs to individual inputs. DP training produces conflicting models even when non-private training results in a single optimal model. Thus, in addition to decreased accuracy, DP-ensuring training methods also incur an arbitrariness cost by exacerbating predictive multiplicity. We show that the degree of predictive multiplicity varies significantly across individuals and can disproportionately impact certain population groups. Fig. 3.1 illustrates the predictive-multiplicity cost of DP training in a simple synthetic scenario (see Section 3.5 for examples on real-world datasets).

Our main contributions are:

1. We provide the first analysis of the predictive-multiplicity cost of differentially-private training.

2. We analyze a method for estimating the predictive-multiplicity properties of randomized machine-learning algorithms using re-training. We derive the first bound on the sample complexity of estimating predictive multiplicity with this approach. Our bound enables practitioners to determine the number of re-trainings required to estimate the predictive-multiplicity cost of randomized training algorithms up to a desired level of accuracy.

3. We conduct a theoretical analysis of the predictive-multiplicity cost of the output perturbation mechanism [33] used to obtain a differentially-private logistic-regression model. We characterize the exact dependence of predictive multiplicity on the level of privacy for this method.

4. We conduct an empirical study of predictive multiplicity of two practical DP-ensuring learning algorithms: DP-SGD [1] and objective perturbation [33]. We use one synthetic dataset and five real-world datasets in the domains of finance, healthcare, and image classification. Our results confirm that, for these

> mechanisms, increasing the level of privacy invariably increases the level of predictive multiplicity. Moreover, we find that different examples exhibit different levels of predictive multiplicity. In particular, different demographic groups can have different average levels of predictive multiplicity.

In summary, the level of privacy in DP training significantly impacts the level of predictive multiplicity. This, in turn, means that decisions supported by differentially-private models can have an increased level of arbitrariness: a given decision would have been different had we used a different random seed in training, even when all other aspects of training are kept fixed and the optimal non-private model is unique. Before deploying DP-ensuring models in high-stakes situations, we suggest that practitioners quantify predictive multiplicity of these models over salient populations and—if possible to do so without violating privacy—measure predictive multiplicity of individual decisions during model operation. Such audits can help practitioners evaluate whether the increase in privacy threatens the justifiability of decisions, choose whether to enact a decision based on a model's output, and determine whether to deploy a model in the first place.

3.2 Technical Background

3.2.1 Problem Setup and Notation

We study randomized training algorithms $T : \mathbb{D}^n \to \Theta$, which produce a parameter vector of a binary classifier in a randomized way. Thus, given a training dataset, $T(S)$ is a random variable. We denote by $P_{T(S)}$ the *model distribution*, the probability distribution over Θ generated by the random variable $T(S)$.

In general, the source of randomness in the training procedure could include, e.g., random initializations of θ prior to training. However, we consider only those sources which are introduced by the privacy-preserving techniques, as we explain in the next section. Throughout this chapter, the datasets, as well as any input example $x \in \mathbb{X}$, are not random variables but fixed values. The only randomness we consider in our notation is due to the internal randomization of the training procedure $T(\cdot)$.

For instance, denoting by $T(S) = T_{\text{np}}(S) + \xi$ the output-perturbation procedure in Eq. (1.25), we treat $T(S)$ as a random variable over the randomness of the injected noise ξ. Other methods to achieve DP such as objective perturbation [33] also inject noise as part of training. In those cases, we similarly consider $T(S)$ as a random variable over such injected noise, and treat all other aspects of training such as pre-training initialization as fixed.

3.2.2 Predictive Multiplicity

Predictive multiplicity occurs when multiple classification models achieve comparable average accuracy yet produce conflicting predictions on a given example [124]. To quantify predictive multiplicity in randomized training, we need to measure dissimilarity of predictions among the models sampled from the probability distribution $P_{T(S)}$ induced by differentially-private training. For this, we use a definition of *disagreement* which has appeared in different forms in [18, 59, 124].

Definition 3.1 (Disagreement). For a given fixed input example $x \in \mathbb{X}$, we define the disagreement $\mu(x)$ as:

$$\mu(x) \triangleq 2 \Pr_{\theta, \theta' \sim P_{T(S)}} [f_\theta(x) \neq f_{\theta'}(x)]. \tag{3.1}$$

In the above definition, $\theta, \theta' \sim P_{T(S)}$ denotes two models sampled independently from $P_{T(S)}$. We use a scaling factor of two in order to ensure that $\mu(x)$ is in the $[0, 1]$ range for the ease of interpretation. A disagreement value $\mu(x) \approx 1$ indicates that the prediction for x is approximately equal to an unbiased coin flip. Moreover, a disagreement $\mu(x) \approx 0$ implies that, with high probability, the prediction for x does not significantly change if two models are independently sampled from $P_{T(S)}$ (i.e., by re-training a model twice with different random seeds).

In the literature, a commonly studied source of variance of outcomes of training algorithms is from re-sampling of the dataset S, usually under the assumption that it is an i.i.d. sample from some data distribution. We do not study variance arising from dataset re-sampling, and are only interested in the predictive-multiplicity properties of the randomized training procedure $T(\cdot)$ itself. Thus, we *fix* both the dataset S used in training and the input example x for which we compute the level of predictive multiplicity, and make sure that the randomness is only due to internal randomization of the training procedure $T(\cdot)$.

When evaluating dissimilarity across models, many prior works that study predictive multiplicity (e.g., [85, 124, 157, 182]) only consider models that surpass a certain accuracy threshold. Although conditioning on model accuracy is theoretically valid, it can bring about confusion in the context of private learning, as in practice such conditioning would demand special mechanisms in order to satisfy DP (see, e.g., [136]). In particular, first applying a DP training method that guarantees an (ϵ, δ)-level of privacy, and then selecting or discarding the resulting model based on accuracy, would result in models that violate the initial (ϵ, δ)-DP guarantees. We note, however, that our results and experiments involving estimation of predictive multiplicity in Sections 3.4 and 3.5 extend to the case in which we add additional conditioning on top of model distribution $P_{T(S)}$ to control for accuracy.

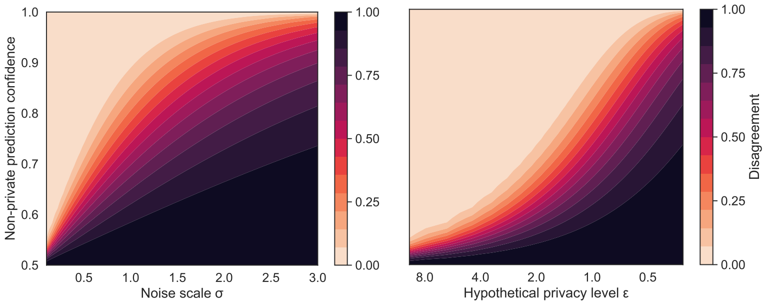


Figure 3.2: **The noise scale in output perturbation mechanisms increases predictive multiplicity for examples which do not attain high non-private prediction confidence.** On the left, the x axis shows the noise scale used for output perturbation (higher values of σ correspond to better privacy). The noise scale corresponds to different levels of privacy depending on the sensitivity of the non-private training algorithm and the δ parameter (see Section 1.3). On the right, the x axis (logarithmic scale) shows a possible level of privacy ϵ for $\delta = 10^{-5}$, assuming that the non-private training algorithm has sensitivity of $C = 0.2$. The y axis shows the hypothetical prediction confidence for a given example. The color intensity shows the level of disagreement (darker means higher disagreement).

Before proceeding with our analyses of disagreement, we first state a simple yet useful relation between disagreement and statistical variance. Observe that for a given input x, the output prediction $f_\theta(x)$ is a random variable over the randomness of the training procedure $\theta \sim P_{T(S)}$. As we assume that the decisions are binary, and training runs are independent, we have that $f_\theta(x) \sim \mathrm{Bernoulli}(p_x)$ for some input-specific parameter p_x. Having noted this fact, we show that disagreement, defined in Eq. (3.1), can be expressed as a continuous transformation of p_x:

Proposition 3.2.1. For binary classifiers, disagreement for a given example $x \in \mathbb{X}$ is proportional to variance of decisions over the distribution of models generated by the training algorithm:

$$\mu(x) = 4 \operatorname{Var}_{\theta \sim P_{T(S)}}(f_\theta(x)) = 4 p_x \, (1 - p_x). \tag{3.2}$$

We provide the proof of this and all the following formal statements in Appendix A.2. Additionally, in Appendix C.2, we provide an analysis using an alternative measure of predictive multiplicity.

3.3 Predictive Multiplicity of Output Perturbation

To demonstrate how DP training can lead to an increase in predictive multiplicity, we theoretically analyze the multiplicity properties of the output-perturbation mechanism described in Section 1.3.

Following Chaudhuri et al. [33] and Wu et al. [190], we study the case of logistic regression. In a logistic-regression model parameterized by vector $\theta \in \mathbb{R}^d$, we compute the confidence score for an input $x \in \mathbb{X} \subseteq \mathbb{R}^d$ as $h_\theta(x) = \mathsf{sigmoid}(\theta^\intercal x)$, where

$$\mathsf{sigmoid}(t) \triangleq \frac{1}{1 + \exp(-t)}. \tag{3.3}$$

Recall that the classifier's prediction is obtained by applying a threshold to the confidence score by Eq. (1.13), in this case as $f_\theta(x) = \mathbb{1}[\mathsf{sigmoid}(\theta^\intercal x) > 0.5]$. Note that the quantity $\theta^\intercal x$ is interchangeable with confidence, as one can be obtained from the other using an invertible transformation. We show the exact relationship between disagreement and the scale of noise σ in this setting:

Proposition 3.3.1. Let $\theta_{\mathsf{np}} = T_{\mathsf{np}}(S)$ be a non-private parameter vector of a logistic-regression model. Suppose that the privatized θ_{priv} is obtained using Gaussian noise of scale σ as in Eq. (1.25). Then, the disagreement of a private logistic-regression model parameterized by θ_{priv} is:

$$\mu(x) = 4\, p_x(1 - p_x), \text{where } p_x = \Phi\left(\frac{\theta_{\mathrm{np}}^\intercal\, x}{\|x\| \cdot \sigma}\right). \tag{3.4}$$

We visualize the relationship in Fig. 3.2, assuming the input space is normalized so that $\|x\| = 1$. There are two main takeaways from this result. First, disagreement is high when the level of privacy is high. Second, the level of multiplicity is unevenly distributed across input examples. This is because the exact relationship between multiplicity and privacy also depends on the confidence of the non-private model, $\theta_{\mathsf{np}}^\intercal\, x$, with lower-confidence examples generally having higher multiplicity in this setting. We note that, in this illustration, the simple relationship between confidence and predictive multiplicity is an artifact of normalized features, i.e., $\|x\| = 1$. In general, examples with high-confidence predictions can display high predictive multiplicity after DP-ensuring training, as illustrated in Section 3.5.2.

Other methods for DP training, such as gradient perturbation [1], are not as straightforward to analyze theoretically. In the next sections, we study predictive multiplicity of these algorithms using a Monte-Carlo method.

3.4 Measuring Predictive Multiplicity of Randomized Algorithms

Theoretically characterizing predictive multiplicity of DP algorithms beyond the output-perturbation mechanism and for more complex model classes is a challenging problem (see, e.g. [85, Section 4]). For instance, the accuracy and generalization behavior of the DP-SGD algorithm [1] used for DP training of neural networks is an active area of research (e.g., [176]). Even in simpler model classes, where training amounts to solving a convex optimization problem (e.g., support vector machines), DP mechanisms such as objective perturbation [33] display a complex interplay between privacy, accuracy, and distortion of model parameters.

For these theoretically intractable cases, we adopt a simple Monte-Carlo strategy [17, 59]: Train multiple models on the same dataset with different randomization seeds, and compute statistics of the outputs of these models. Note that this procedure does not preserve differential privacy, which we discuss in more detail in Section 3.7.2.

In this section, we formalize this simple and intuitive approach, and provide the first sample complexity bound for estimating predictive multiplicity. Our bound has a closed-form expression, so a practitioner can use it to determine how many re-trainings are required to estimate predictive multiplicity up to a given approximation error.

At first, re-training might appear as a blunt approach for analyzing predictive multiplicity in DP. Our results indicate that this is not the case. Surprisingly, we prove that, if one wants to estimate disagreement in Eq. (3.1) for k input examples, the number of required re-trainings increases *logarithmically* in k. This result demonstrates that re-training can be an effective strategy to estimate predictive multiplicity regardless of the intricacies of a specific DP mechanism, and that a moderate number of re-trainings is sufficient to estimate disagreement for a large number of examples.

Recall that, according to Proposition 3.2.1, disagreement of an example x is proportional to the variance of outputs within the model distribution $P_{T(S)}$. We use this connection to provide an unbiased estimator for disagreement.

Proposition 3.4.1. Suppose we have m models sampled from the model distribution: $\theta_1, \theta_2, \ldots, \theta_m \sim P_{T(S)}$. Then, the following expression is an unbiased estimator for disagreement $\mu(x)$ for a single example $x \in \mathbb{X}$:

$$\hat{\mu}(x) \triangleq 4 \frac{m}{m-1} \hat{p}_x (1 - \hat{p}_x), \tag{3.5}$$

where $\hat{p}_x = \frac{1}{m} \sum_{i=1}^{m} f_{\theta_i}(x)$ is the sample mean of $f_\theta(x)$.

How many models $\theta_1, \theta_2, \ldots, \theta_m$ do we need to sample in order to estimate disagreement? To answer this, we provide an upper bound on estimation accuracy given the number of samples from the model distribution, as well as a bound on the number of samples required for a given level of estimation accuracy.

Proposition 3.4.2. For m models sampled from the model distribution, $\theta_1, \theta_2, \ldots, \theta_m \sim P_{T(S)}$, with probability at least $1 - \rho$, for $\rho \in (0, 1]$ the additive estimation error $\alpha \triangleq |\hat{\mu}(x) - \mu(x)|$ satisfies:

$$\alpha \leq \frac{1}{(m-1)} + 4\frac{m}{m-1}\sqrt{\frac{\log(2/\rho)}{2m}}\left(1 + \sqrt{\frac{\log(2/\rho)}{2m}}\right). \tag{3.6}$$

For example, this bound yields that 5,000 re-trainings result in the estimation error of at most 0.08 with probability 95%. In Appendix A.2.2, we provide a closed-form expression for computing the number of samples m required to achieve a given error level α. We also provide a visualization of the bound in Fig. C.3a (Appendix).

In practice, one might need to estimate disagreement for multiple examples, e.g., to compute average disagreement over a test dataset. When doing so naively, the re-training costs could mount to infeasible levels if we assume that each estimation requires the same number of models, m, for each input example. In contrast, we show that in such cases sample complexity grows only logarithmically.

Proposition 3.4.3. Let $x_1, x_2, \ldots, x_k \in \mathbb{X}$. If $\theta_1, \theta_2, \ldots, \theta_m \sim P_{T(S)}$ are i.i.d. samples from the model distribution, then with probability at least $1 - \rho$, for $\rho \in (0, 1]$ the maximum additive error satisfies:

$$\begin{aligned}\max_{j \in 1, \ldots, k} |\mu(x_j) - \hat{\mu}(x_j)| \leq \frac{1}{(m-1)} &+ \\ &+ \frac{4m}{m-1}\sqrt{\frac{\log(2k/\rho)}{2m}}\left(1 + \sqrt{\frac{\log(2k/\rho)}{2m}}\right).\end{aligned} \tag{3.7}$$

This positive result shows that auditing models for predictive multiplicity for large populations and datasets is practical, as the sample complexity grows slowly in the number of examples.

3.5 Empirical Studies

In this section, we empirically explore the predictive multiplicity of DP algorithms. We use a low-dimensional synthetic dataset in order to visualize the level of multiplicity across the input space. To study predictive-multiplicity effects in realistic settings, we use real-world tabular datasets representative of high-stakes domains, namely lending and healthcare, and one image dataset.

3.5.1 Experimental Setup

Datasets and Tasks. We use the following datasets:

- A **Synthetic** dataset containing data belonging to two classes with class-conditional distributions $X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $X_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$, respectively. We set the distribution parameters to be:

$$\begin{aligned} \mu_0 &= [1, 1], & \Sigma_0 &= \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}, \\ \mu_1 &= [-1, -1], & \Sigma_1 &= \begin{pmatrix} 1 & 1/10 \\ 1/10 & 1 \end{pmatrix}. \end{aligned} \tag{3.8}$$

  The classes in this synthetic dataset are well-separable by a linear model (see Fig. 3.1)

- Credit Approval tabular dataset (**Credit**). The task is to predict whether a credit card application should be approved or rejected based on several attributes which describe the application and the applicant.
- Contraceptive Method Choice tabular dataset (**Contraception**) based on 1987 National Indonesia Contraceptive Prevalence Survey. The task is to predict the choice of a contraception method based on demographic and socio-economic characteristics of a married couple.
- Mammographic Mass tabular dataset (**Mammography**) collected at the Institute of Radiology of the University Erlangen-Nuremberg in 2003 − 2006. The task is to predict whether a screened tumor is malignant or benign based on several clinical attributes.
- **Dermatology** tabular dataset. The task is to predict a dermatological disease based on a set of clinical and histopathological attributes.
- **CIFAR-10** [103], an *image* dataset of pictures labeled as one of ten classes. The task is to predict the class.

We take the realistic tabular datasets (Credit, Contraception, Mammography, and Dermatology) from the University of California Irvine Machine Learning (UCIML) dataset repository [52]. We provide a summary of the dataset characteristics in Table 3.1. In Appendix C.2, we provide additional details about processing of the datasets.

Table 3.1: Summary of datasets used in our experimental evaluations.

Dataset	Size	Number of features	Train size	Test size
Synthetic	∞	2	2000	20,000
Credit	653	46	489	164
Contraception	1,473	9	1,104	369
Mammography	830	5	622	208
Dermatology	358	34	268	90
CIFAR-10	60,000	$32 \times 32 \times 3$	50,000	10,000

For the synthetic dataset, we obtain the training dataset by sampling 1,000 examples from each of the distributions. In order to have precise estimates of population accuracy, we sample a larger test dataset of 20,000 examples. For tabular datasets, we use a random 75% subset for training, and use the rest as a held-out test dataset for model evaluations. For CIFAR-10, we use the default 50K/10K train-test split.

Models and training algorithms. For the synthetic and tabular datasets, we use logistic regression with objective perturbation [33]. For the image dataset, we train a convolutional neural network on ScatterNet features [134] using DP-SGD [1], following the approach by Tramer and Boneh [169]. We provide more details in Appendix C.2.

Metrics. The goal of our experiments is to quantify predictive multiplicity and explain the factors which impact it. For all settings, we measure disagreement to capture the dissimilarity of predictions, and predictive performance of the models to quantify the effect of performance on multiplicity. Concretely, we measure:

- **Disagreement** for examples on a test dataset, computed using the unbiased estimator in Section 3.4. As this disagreement metric is tailored to binary classification, we use a special procedure for the ten-class task on CIFAR-10: we treat each multi-class classifier as ten binary classifiers, and we report average disagreement across those ten per-class classifiers. Additionally, in Appendix C.2, we also report predictive multiplicity in terms of confidence scores instead of predictions following the recent approach by Watson-Daniels et al. [182].
- **Performance** on a test dataset. For tabular datasets, we report the standard area under the ROC Curve (AUC for short). For CIFAR-10, we report accuracy.

Experiment outline. For a given dataset and a value of the privacy parameter ϵ, we train multiple models on exactly the same data *with different randomization seeds.*

For the synthetic and tabular datasets, we use several values of ϵ between 0.5 (which provides a good guaranteed level of privacy [see, e.g. 189, Section 4]) and 2.5, with $\delta = 0$. For each value of ϵ we train $m = 5{,}000$ models. For CIFAR-10, we train $m = 50$ neural-network models because of computational constraints. We use DP-SGD parameters that provide privacy guarantees from $\epsilon \approx 2$ to $\epsilon \approx 7$ at the standard choice of $\delta = 10^{-5}$.

3.5.2 Predictive Multiplicity and Privacy

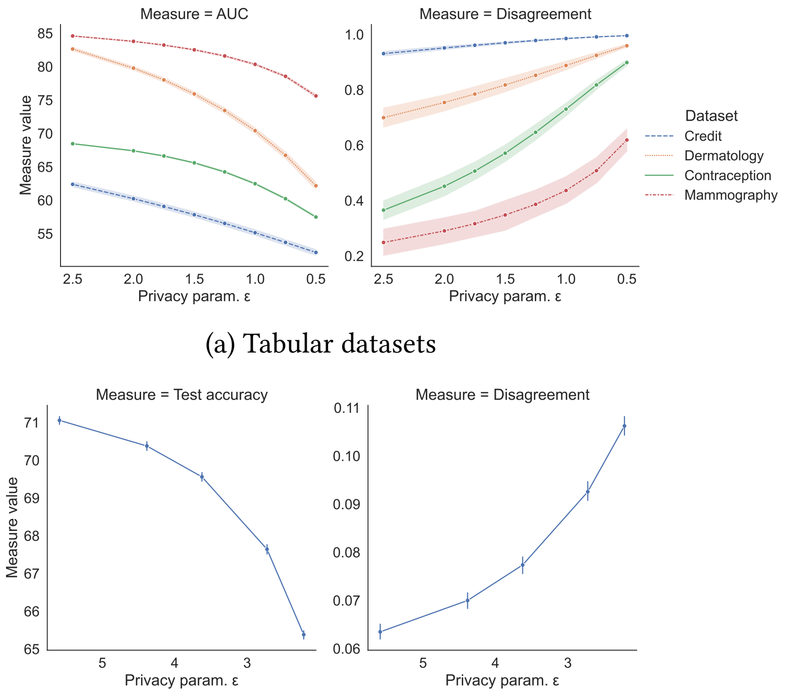
First, we empirically study how multiplicity evolves with increasing privacy. In Fig. 3.1, we visualize the two-dimensional synthetic examples and their disagreement for different privacy levels. As privacy increases, so do the areas for which model decisions exhibit high disagreement (darker areas). Although the regions with higher disagreement correlate with model confidence and accuracy, the level of privacy contributes significantly. For instance, some points which are relatively far from the decision boundary, which means they are confidently classified as either class, can nevertheless have high predictive multiplicity.

Fig. 3.3 shows the experimental results for our tabular datasets and CIFAR-10. On the left side, we show the relationship between the privacy level and performance. On the right, between the privacy level and disagreement. As with the theoretical analysis and the results on synthetic data, we can clearly see that models with higher level of privacy (low ϵ) invariably exhibit higher predictive multiplicity. Notably, even for datasets such as Mammography and CIFAR-10 for which *average* disagreement is relatively low, there exist examples whose disagreement is 100%. See Table C.1 in the Appendix for detailed information on the distribution of the disagreement values across the test data.

Implications. The increase in the privacy level results in making more decisions which are partially or fully explained by randomness in training. Let us give an example with a concrete data record from the Mammography dataset representing a 56-year-old patient labeled as having a malignant tumor. Classifiers with low level of privacy $\epsilon = 2.5$ predict the correct malignant class for this individual most of the time (approx. 55% disagreement). If we set the level of privacy to the high $\epsilon = 0.5$, this record is classified close to 42% of the time as benign, and 58% of the time as malignant (approx. 97% disagreement). Thus, if one were to use a model with the high level of privacy to inform treatment of this patient, the model's decision would have been close in its utility to a coin flip.

3.5.3 What Causes the Increase in Predictive Multiplicity?

In the previous section, we showed that the increase in privacy causes an increase in predictive multiplicity. It is not clear, however, what is the exact mechanism through which DP impacts predictive multiplicity. Hypothetically, the contribution to multiplicity could be through two pathways:



(a) Tabular datasets

(b) Image dataset (CIFAR-10)

Figure 3.3: **Increasing the level of privacy increases the level of predictive multiplicity in real-world datasets.** For all plots, the x axis shows the level of privacy (ϵ, lower value is more privacy). The plots on the left shows the performance level (AUC for tabular datasets, and accuracy for CIFAR-10). The error bands/bars on the left side are 95% confidence intervals (CI) over the models in the model distribution. The plots on the right show the degree of disagreement across $m = 5{,}000$ models in the case of tabular datasets, and across $m = 50$ models in the case of CIFAR-10. The error bands/bars on the right side are 95% CI over the examples in a test dataset. Although average disagreement might be relatively low for some datasets such as Mammography and CIFAR-10, there exist examples for which disagreement is 100% (see Table C.1 in the Appendix).

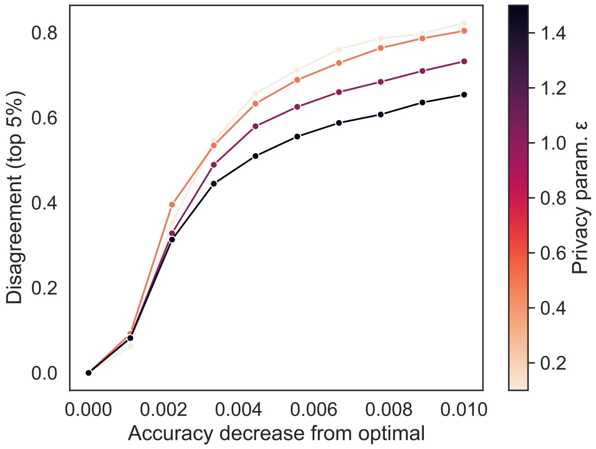


Figure 3.4: **Models achieving a similar level of accuracy can have different levels of predictive multiplicity.** The plot shows the top 5% percentile of disagreement on the synthetic test dataset for all models which attain at least certain level of accuracy, for different values of the privacy parameter (ϵ, lower value is more privacy). The x axis shows the deviation of accuracy from that of an optimal non-private model, with 0 being equal to the accuracy of the optimal non-private model. As even such a small decrease in accuracy as 0.01 can see disagreement rise from 0 to 0.8 for some examples, this result suggests that the change in the level of privacy on its own can cause a big change in disagreement.

(1) *Direct:* The increase in predictive multiplicity is the result of the variability in the learning process stemming from randomization, regardless of the performance decrease.

(2) *Indirect*: The increase in predictive multiplicity is the result of the decrease in performance.

These two options are not mutually exclusive, and it is possible that both play a role. In both cases, the desire for a given level of privacy—which determines the degree of randomization added during training—is ultimately the cause of the increase in multiplicity. Nevertheless, how randomization contributes to the increase has practical implications: If our results are explained by pathway (2), we should be able to reduce the impact of privacy on predictive multiplicity by designing algorithms which achieve better accuracy at the same privacy level.

For output perturbation, our analysis in Section 3.3 shows that multiplicity is directly caused by randomization—pathway (1)—as only the privacy level, confidence, and the norm of a predicted example impact disagreement. Therefore, performance does not have a direct impact on predictive multiplicity in output perturbation.

In Fig. 3.4, to quantify the impact of performance on predictive multiplicity for the case of objective perturbation, we show the top 5% disagreement values for varying levels of accuracy on the synthetic dataset. We use the synthetic dataset to ensure that test accuracy estimates are reliable, as we have a large test dataset in this case. We see that, for a given level of accuracy, different privacy parameters can result in different disagreement. This suggests that randomization caused by DP training *can have a direct effect* on predictive multiplicity, so we observe pathway (1).

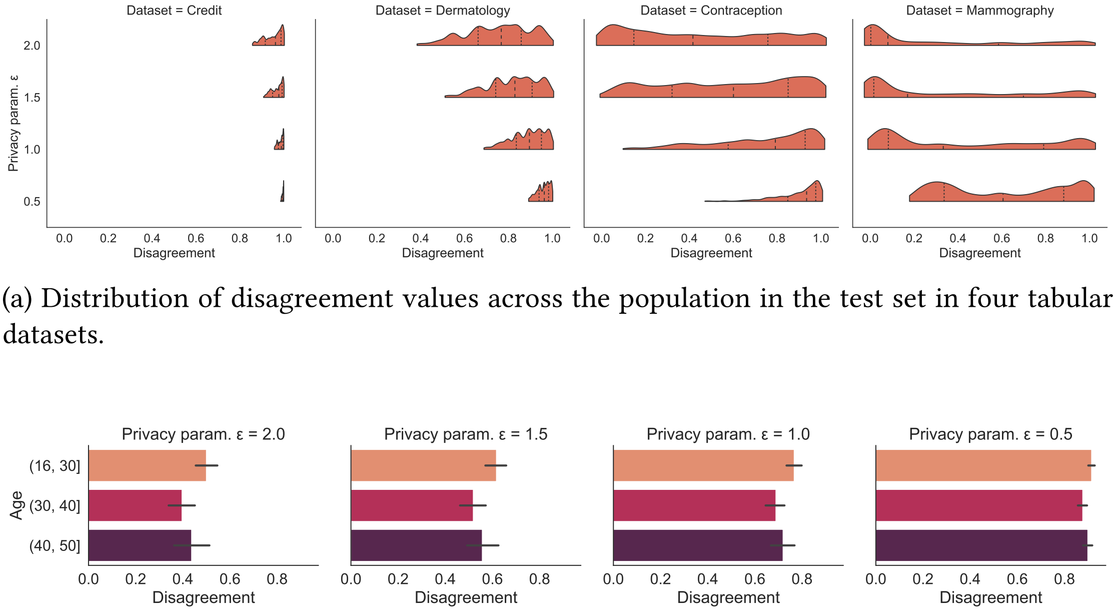
Implications. This observation indicates that there exist cases for which improving accuracy of a DP-ensuring algorithm at a given privacy level *will not* necessarily lower predictive multiplicity.

3.5.4 Disparities in Predictive Multiplicity

The visualizations in Fig. 3.1 show that different examples can exhibit highly varying levels of predictive multiplicity. This observation holds for real-world datasets too. Fig. 3.5a shows the distributions of the disagreement values across the population of examples in the test data for tabular datasets. For example, for lower privacy levels (high ϵ) on the Contraception dataset, there are groups of individuals with different values of predictive multiplicity. As the level of privacy increases (low ϵ), the disagreement tends to concentrate around 1, with decisions for a majority of examples largely explained by randomness in training.

Next, we verify if the differences in the level of disagreement also exist across demographic groups. In Fig. 3.5b, we show average disagreement across points from three different age groups in the Contraception dataset. As before, for low levels of privacy (high ϵ) we see more disparity in disagreement. The disparities even out as we increase the privacy level (low ϵ), with groups having average disagreement closer to 1. Thus, disagreement is not only unevenly distributed across individuals, but across salient demographic groups.

Implications. As some groups and individuals can have higher predictive multiplicity than others, evaluations of training algorithms in terms of their predictive multiplicity must account for such disparities. For instance, our experiments on the Contraception dataset (in Fig. 3.5b) show that, for different privacy levels, decisions for individuals in the 16–30 age bracket exhibit higher predictive multiplicity than of patients between 30 and 40 years old. Predictions for individuals under 30, therefore, systematically exhibit more dependence on randomness in training than on the relevant features for prediction. This highlights the need to conduct disaggregated evaluations as opposed to only evaluating average disagreement on whole datasets.



(a) Distribution of disagreement values across the population in the test set in four tabular datasets.

(b) Group-level disparities in disagreement values on the Contraception dataset. Error bars are 95% confidence intervals over the disagreement values in each subgroup.

Figure 3.5: **The level of predictive multiplicity varies from one example to another, and across population groups. As the level of privacy grows, more predictions exhibit similarly high disagreement.**

3.6 Related Work

Rashomon Effect and Predictive Multiplicity. The Rashomon effect, observed and termed by Breiman [21], describes the phenomenon where a multitude of distinct models achieve similar average loss. The Rashomon effect occurs even for simple models such as linear regression, decision trees, and shallow neural networks [7]. When no privacy constraints are present, predictive multiplicity can be viewed as a facet of the Rashomon effect in classification tasks, where similarly-accurate models produce conflicting outputs. One of the main challenges in studying predictive multiplicity is measuring it. Semenova et al. [157] proposed the Rashomon ratio to measure the Rashomon effect and used a Monte Carlo technique to sample decision tree models for estimation. Marx et al. [124] quantified predictive multiplicity using optimization formulations to find the worst-case disagreement among all candidate models while controlling for accuracy. Recently, Hsu and Calmon [85], Watson-Daniels et al. [182] proposed other metrics for quantifying predictive multiplicity: Rashomon capacity and viable prediction range. Black et al. [18] proposed measures of predictive multiplicity which are applicable to randomized learning. Our Proposition 3.4.3 complements the prior work by providing a closed-form expression for sample complexity of estimating predictive multiplicity which arises due to randomness in training.

Side Effects of Differential Privacy. To the best of our knowledge, our work is the first one to study the properties of DP training in terms of predictive multiplicity. Multiple works, however, have studied other unintended consequences of private learning. In particular, a number of works [8, 66, 154] show that DP training comes at a cost of decreased performance for groups which are under-represented in the data. Relatedly, Cummings et al. [45] show that DP training is incompatible with some notions of algorithmic fairness.

3.7 Discussion

Our theoretical and empirical results show that training with DP and, more broadly, applying randomization in training increases predictive multiplicity. We demonstrated that higher privacy levels result in higher multiplicity. If a training algorithm exhibits high predictive multiplicity for a given input example, the decisions supported by a model's output for this example lose their justifiability: these decisions depend on the randomness used in training rather than on relevant properties or features of this example. The connection between privacy in learning and decision arbitrariness might not be obvious to practitioners. This lack of awareness is potentially damaging in high-stakes settings (e.g., medical diagnostics, lending, education), where decisions of significant—and potentially life-changing—consequence could be significantly influenced by randomness used to ensure privacy.

In this concluding section, we discuss whether predictive multiplicity is indeed a valid concern for DP-ensuring algorithms, and outline a path forward.

3.7.1 Can the Increase in Predictive Multiplicity be Beneficial?

Despite the harms of arbitrariness, one might argue that multiplicity can, in some cases, be beneficial.

Opportunities for satisfying desirable properties beyond accuracy? Black et al. [18] and Semenova et al. [157] argue that multiplicity presents a valuable opportunity. In non-private training, the existence of many models that achieve comparable accuracy creates an opportunity for selecting a model which satisfies both an acceptable accuracy level and other useful properties beyond performance, such as fairness [40], interpretability [64], or generalizability [157]. In order to leverage this opportunity, one needs to deliberately steer training towards the model which satisfies desirable properties beyond accuracy, or search the "Rashomon set" of good models [64]. However, with randomization alone (e.g., adding Gaussian noise to gradients in training), model designers cannot steer training without compromising DP guarantees, and can only arrive at a model which satisfies additional desirable properties by chance. Thus, this positive side of the multiplicity phenomenon is not necessarily present in DP-ensuring training.

It is an open problem to find whether specially-crafted noise distributions or post-processing techniques could be designed to provide the same level of privacy as the standard approaches, and at the same time attain additional useful properties such as algorithmic fairness.

Predictive multiplicity is individually fair? Individual fairness [55] is a formalization of the "treat like alike" principle: an individually fair classifier makes similar decisions for individuals who are thought to be similar. A way to formally satisfy individual fairness is, in fact, through randomization of decisions. This could lead to an argument that predictive multiplicity is individually fair. For instance, suppose that a predictive model used to assist with hiring decisions is applied to several individuals who are all equally qualified to get the job. Consider two possible decision rules for selecting the candidate to hire with different multiplicity levels. The first rule has high multiplicity: produce a random decision. The second rule has low multiplicity: select a candidate based on lexicographic order. As the second decision rule results in a breach of individual fairness and, possibly, a systemic exclusion of some candidates, the first rule with high multiplicity seems preferable.

This argument, however, only holds if there is randomness *at the prediction stage.* This is not the case for standard DP-ensuring algorithms such as the ones we study. DP training produces one deterministic classifier that is used for all predictions. Thus, once training is done, there is no randomization of decisions as in the example above. Thus, the decisions due to such DP-ensuring models are no different than arbitrary rules such as selection based on lexicographic ordering.

Overcoming the algorithmic Leviathan? Creel and Hellman [42] consider a setting where different decision-making systems which have high impact on an individual's livelihood, e.g., credit scoring systems from competing bureaus in the USA [39], are trained in ways that lead to all of them outputting the same decisions. This *algorithmic monoculture* would completely remove the possibility of accessing resources for some individuals, as turning to a competing decision-maker would not change the outcome. In this case, Creel and Hellman argue that high predictive multiplicity could be a desirable property as it enables to access resources across the decision-makers.

In some high-stakes settings, such as healthcare, an algorithmic monoculture might *not* pose a concern. Indeed, one would wish that predictive models used as a part of a diagnostic procedure for a disease output a consistent decision so that patients can be treated (or not treated) as needed. In this scenario, in fact, predictive multiplicity could potentially harm patients by either delaying a patient's treatment, or recommending a treatment when the patient is healthy. In such settings, the positive impact of predictive multiplicity in avoiding an algorithmic Leviathan loses meaning.

Regardless of whether algorithmic monoculture is a legitimate concern or not for a given application, it is helpful for model designers and decision subjects to be informed of the level of predictive multiplicity, whether to gauge the likelihood of recourse, or brace for the arbitrariness of decisions.

3.7.2 Open Problems

Reporting mutiplicity. Potential mitigations of the harms of predictive multiplicity could be to abstain from outputting a prediction with high multiplicity, or to communicate the magnitude of multiplicity to the stakeholders. Doing so is challenging: any sort of communication of disagreement values could partially reveal information about the privacy-sensitive training data and break DP guarantees. Consider, as before, the setting of using a predictive model to assist in a medical diagnosis. Whether a model abstains from predictions or outputs them along with disagreement estimates, there is a certain amount of information leakage about the training data to doctors. If the disagreement estimates are computed on privacy-sensitive data and are used without appropriate privatization—whether published or used to decide on abstention—they can reveal information about the data. To address this issue, one could use privacy-preserving technologies such as DP to abstain from making a prediction based on a high disagreement value or report the disagreement estimate in a privacy-preserving way. Studying whether effective privatization of disagreement computations is possible is an open problem for future work.

General characterization of the predictive-multiplicity costs of DP. We have theoretically characterized the predictive-multiplicity behavior of the output-perturbation mechanism as applied to logistic regression. Doing so for other mechanisms and model families is a non-trivial undertaking. In this work, we resort to empirical measurement with re-training. An open problem is finding whether we can characterize these behaviors for a wider range of model families, mechanisms, or even for any general mechanism which satisfies DP.

3.7.3 Recommendations Moving Forward

As discussed in the previous sections, existing techniques do not enable model designers to eliminate, or even mitigate, the implications of predictive multiplicity when using DP-ensuring models. We have pointed out which open problems would need to be solved in order to reduce the impact of predictive multiplicity in high-stakes privacy-sensitive scenarios. Until DP mechanisms that mitigate multiplicity become available, the negative effects of multiplicity can only be countered by *auditing for multiplicity* prior to deployment. Therefore, in order to understand the impact of privacy on the justifiability of model decisions, model designers should directly measure predictive multiplicity when using DP training, e.g., using the methods we introduce in Section 3.4. If at the desired level of privacy the training algorithm exhibits high predictive multiplicity (either in general or for certain populations), model designers should carefully consider whether the use of such models is justified in the first place.

Chapter 4

Unequal Access to Privacy

This chapter is based on a peer-reviewed article entitled “Disparate Vulnerability to Membership Inference Attacks” [106] by Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso, published in the 2022 Proceedings on Privacy Enhancing Technologies (PoPETS).

As detailed in Chapter 2, the concept of distributional generalization (DG) was initially introduced in the article on which the present chapter is based. In this thesis, however, the concept is introduced in Chapter 2 for clarity of exposition.

4.1 Introduction

Membership Inference Attacks (MIAs), in which an adversary aims to determine whether an example is part of the training set, are one of the main privacy attacks against machine-learning (ML) models. Since they were first described in the context of ML [161], many works have studied the potential of these attacks under diverse circumstances [90, 119, 131], as well as the causes and limits of these attacks [62, 112, 194]. In both empirical and theoretical approaches researchers focus on the *average* MIA success across examples. However, there is empirical evidence that the vulnerability to MIAs is not always evenly distributed: it can differ across target classes [161], it can be more effective against some individuals [119], and it can vary across subgroups [31]. These results imply that average-based studies can overestimate the privacy for some individuals [61].

In this chapter, we present the first theoretical analysis of the disparate vulnerability to MIAs across populations and subgroups. Our contributions encompass several key aspects. First, we introduce a novel characterization of vulnerability to MIAs, establishing a necessary and sufficient condition for these attacks to succeed. We find out that such necessary and sufficient condition is, in fact, poor distributional generalization, described in Chapter 2. Specifically, vulnerability arises when the distribution of a model's properties, such as loss or outputs, differs between examples from the training dataset and those outside it.

We extend this analysis to explore disparate vulnerability across subgroups, introducing the first formal examination of this phenomenon. We provide insights into necessary and sufficient conditions for preventing MIAs while considering subgroup vulnerability and disparate vulnerability. Additionally, we address the challenges associated with estimating the magnitude of disparate vulnerability when subgroups are small. To tackle this issue, we present a statistical framework and methods for estimating disparate vulnerability and determining its significance. Our study reveals that not all vulnerability estimation mechanisms are suitable for analyzing subgroups, and we discuss the implications of these difficulties for regulatory compliance.

Our contributions are the following:

- We introduce a novel characterization of the vulnerability to MIAs, which provides a *necessary and sufficient* condition for these attacks to succeed: lack of *distributional generalization.* Vulnerability to MIA arises when the *distribution* of a model's property (e.g., loss, or outputs) is different for samples in and out of the training dataset. This result complements previous studies that demonstrated the lack of standard generalization (i.e., overfitting) to be a sufficient but not necessary condition for vulnerability to MIAs [119, 194].

- We introduce the first formal analysis of disparate vulnerability and extend our results on necessary and sufficient conditions for preventing MIAs to subgroup vulnerability and disparate vulnerability.
- We show that estimating the magnitude of the disparate vulnerability is non-trivial when subgroups are small. We provide a statistical framework and methods to estimate disparate vulnerability and its significance. We show that not all vulnerability estimation mechanisms used in prior work are adequate for subgroups. We discuss the implications of these difficulties for regulation compliance.
- We prove that satisfying algorithmic-fairness constraints can decrease disparate vulnerability to limited classes of attackers. We also show that training with differential privacy bounds the magnitude of the disparate vulnerability.
- We empirically evaluate disparate vulnerability on synthetic and on real-world datasets, demonstrating that disparate vulnerability exists in realistic models with high statistical significance.
- We discuss the importance of disaggregating privacy measurements when evaluating the legal implications of privacy attacks. In particular, the importance of studying the consequences of privacy attacks for subgroups when analyzing the privacy risks of a deployment, as opposed to studying individual privacy risks [118] that can be dismissed as residual and acceptable.

In summary, this chapter offers a comprehensive analysis of the disparate vulnerability to MIAs, unveiling necessary and sufficient conditions for their success, exploring subgroup vulnerability, and providing statistical frameworks for estimation. By demonstrating the effectiveness of fairness constraints and differential privacy in mitigating vulnerability and presenting empirical evidence, this work contributes insights into the practical implications of privacy attacks. Our emphasis on subgroup analysis underscores the importance of considering privacy risks for different populations, ensuring an inclusive approach to privacy protection.

4.2 Membership Inference Attacks

Membership Inference Attacks (MIAs) are the basic privacy risk against which differentially private training aims to protect (see Section 1.3). The goal of a MIA is to predict whether an example $z \in \mathbb{D}$ is a *member* or a *non-member* of the training dataset S of a given predictive model $f_\theta(\cdot)$. We assume a threat model where a MIA adversary observes the target model's behavior that relates to z, and has information about the data distribution P, training-data sampling, and the training algorithm. We formalize

MIAs using the indistinguishability game by Yeom et al. [194]. We say that the game is between an adversary that aims to conduct a MIA and a *challenger*. Formally, the game $\mathrm{MIA}(\mathcal{A}, T, n, P)$ proceeds as follows:

1. $S \leftarrow P^n; \theta = T(S)$
2. $m \sim \{0, 1\}$
3. if $m = 1$ then $z \sim S$, else $z \sim P$
4. $\hat{m} \leftarrow \mathcal{A}(z; \theta, T, n, P)$
5. return $m = \hat{m}$

In this game, the challenger obtains a dataset S as an i.i.d. sample from the data distribution, and trains a model θ using the training algorithm $T(\cdot)$ (line 1). The challenger then draws a secret m uniformly at random (line 2). The value of m denotes x's $m$embership in S: $m = 1$ if the *challenge example* z is sampled from the training dataset S (line 4), and $m = 0$ if it is sampled from the data distribution P (line 6). As Yeom et al. [194], we assume that the population is large enough that the chance of sampling a member $z \in S$ from P is negligible. Given the challenge example z, the target model θ and its training algorithm $T(\cdot)$, the sampling parameter n, and the distribution of the training data P, the MIA adversary $\mathcal{A}(\cdot)$ makes a guess $\hat{m}$ about the example's membership in S (line 5). We use this formalization as it is common, although there are other ways to formalize MIAs [see, e.g., 87]. Note that the MIA game defines a joint probability distribution over training datasets S, membership labels m, and challenge examples z.

4.2.1 Attack Strategy

As described in the MIA game, the adversary's knowledge is limited to $(z; \theta, T, n, P)$, and their goal is to guess the membership of z. We define a general strategy to perform a membership attack that encompasses several instances of MIA, e.g., [131, 161, 194]. This strategy consists of two phases.

First, the adversary prepares an attack algorithm $\mathtt{Att}_{T,n,P}(\cdot)$ which depends on the target training algorithm $T(\cdot)$, and the data-sampling parameters n and P, e.g., by training a shadow-model attack classifier [161]. We drop the subscripts in $\mathtt{Att}_{T,n,P}$ where the setting is clear from the context.

In the second phase, the adversary extracts *features* $w \in \mathbb{W}$ using the *feature extraction function* $w = \pi(z; \theta)$, describing the target model and the example, and applies the attack algorithm to the extracted features to obtain the membership guess, $\hat{m} \leftarrow \mathtt{Att}_{T,n,P}(w)$. Thus, the guess $\hat{m}$ is obtained by applying the attack algorithm to the extracted features:

$$\mathcal{A}(z; \theta, T, n, P) \triangleq \mathtt{Att}_{T,n,P} \circ \pi(z; \theta)$$

This formalization is flexible: it captures both white-box and black-box adversarial models. For example, the features could be the outputs of the model and the example's label $w = (f_\theta(x), y)$ [161], the model's loss for the challenge example, $w = \ell(z; \theta)$ [194], or the model's gradients as in some white-box attacks [131]. We denote by $\mathcal{A}_\pi$ an adversary that uses features $\pi(z; \theta)$. We drop the subscripts where clear from context.

In this chapter, we distinguish two kinds of adversaries depending on the features they use: *regular* adversaries that do not use subgroup information ($G \notin w$), and *subgroup-aware* adversaries that do use this information ($G \in w$). As described in Chapter 1, we assume that the latter adversary can obtain the subgroup G from the examples z themselves as $G = g(z)$. That is the case for our experiments on real-world data in Section 4.6. Prior work has mainly considered regular adversaries.

Given a feature function $\pi(z; \theta)$ that does not include subgroup information, we denote a feature function that uses subgroup information in addition to features of $\pi(z; \theta)$ as $\pi \circ g$. Therefore, if $\mathcal{A}_\pi$ is a regular adversary, then $\mathcal{A}_{\pi \circ g}$ is a subgroup-aware adversary.

4.2.2 Vulnerability

We introduce the concept of *vulnerability* of an ML model to membership inference attacks (MIAs). Vulnerability measures the success of an adversary against the model. We also introduce worst-case (Bayes) vulnerability, i.e., vulnerability against an information-theoretically optimal adversary.

Vulnerability to MIAs is the normalized advantage [194] of adversary $\mathcal{A}$ over random guessing:

Definition 4.1. We define *vulnerability* to adversary $\mathcal{A}$ as:

$$V(\mathcal{A}) \triangleq 2 \Pr[\mathrm{MIA}(\mathcal{A}, T, n, P) = 1] - 1 \tag{4.1}$$

We also extend the definition to subgroups:

Definition 4.2. Let $G \in \mathbb{G}$ be a subgroup of the population. We define *subgroup vulnerability* to adversary $\mathcal{A}$ as:

$$V_G(\mathcal{A}) \triangleq 2 \Pr[\mathrm{MIA}(\mathcal{A}, T, n, P) = 1 \mid z \in G] - 1.$$

which captures the normalized advantage of a MIA adversary $\mathcal{A}$ for challenge examples coming from a given subgroup G.

Optimal adversaries. We base our analysis on information-theoretically optimal adversaries. The worst-case vulnerability to any adversary that leverages features $\pi(z;\theta)$ is:

$$\max_{\texttt{Att}:\mathbb{W}\mapsto\{0,1\}} V(\texttt{Att} \circ \pi). \tag{4.2}$$

The maximum is achieved by a *Bayes adversary* which uses the following strategy for the attack [36, 150]:

$$\texttt{Att}^*(w) \triangleq \operatorname*{argmax}_{m\in\{0,1\}} \Pr[m \mid w], \tag{4.3}$$

over the randomness of the $\mathrm{MIA}(\mathcal{A}, T, n, P)$. In other words, the Bayes adversary uses a Bayes classifier (see Chapter 1) to perform the attack. We denote the Bayes adversary as $\mathcal{A}^*_\pi \triangleq \texttt{Att}^* \circ \pi$.

Subgroup-aware Bayes adversary. We assume the adversary can know the subgroup G to which each example z belongs. Recall that we refer to this adversary as subgroup-aware. As the vulnerability to the Bayes adversary grows if the adversary has more information about the examples, the worst-case vulnerability to a subgroup-aware adversary is equal or higher compared to a regular adversary:

Proposition 4.2.1. $V(\mathcal{A}^*_{\pi\,\circ\,g}) \geq V(\mathcal{A}^*_\pi)$.

We defer the proof to Appendix A.3.

In our experimental evaluations, we only consider subgroup-aware adversaries as they are guaranteed to attain higher advantage in the worst case.

4.3 Distributional Generalization and Vulnerability to MIAs

An ML model is said to *overfit*, or poorly *generalize*, when its average loss on the training set differs from its loss on new samples from the population. Previous work showed that, while overfitting is an important factor for evaluating MIA [161], it is not necessary for MIA vulnerability [119, 194]. In this section, we aim to find a characterization of MIAs that enables us to determine the necessary and sufficient conditions for models to be vulnerable to these attacks.

Fig. 4.1 illustrates with an example why the absence of standard overfitting does not, in general, prevent MIAs. The figure shows a model's loss values on its training and test data. The standard, average-based definition of overfitting cannot distinguish between the two distributions; but an adversary potentially can, and the model can be vulnerable to MIAs. In order to establish the necessary and sufficient conditions for models to be vulnerable to MIAs, we use the extended notion of generalization that

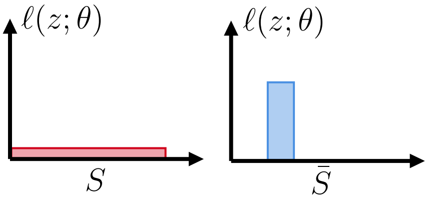


Figure 4.1: Loss values of a model θ on train data S (left) and test data $\bar{S}$ (right). According to the standard notion of generalization, this model does not overfit as the average loss (area) on training and test data is identical. Some population individuals, however, are more penalized on the test data. This discrepancy is captured by *distributional* generalization.

goes beyond comparing the average loss on train and test data introduced in Chapter 2, distributional generalization.

Specifically, let us restate the property-distributional generalization from Definition 2.2 in terms of its violation. We make use of the two probability distributions of examples and models on train and test data, respectively:

$$\begin{aligned} P_1 &\triangleq (z; T(S)), \text{ where } S \sim P^n, z \sim S \\ P_0 &\triangleq (z; T(S)), \text{ where } S \sim P^n, z \sim P \end{aligned} \tag{4.4}$$

Consider a *property*, any function that takes as input a model and an example: $\pi(z;\theta)$, and returns a numeric vector in $\mathbb{R}^k$. A property function could be, for instance, a loss function, the gradient, or the prediction from the model.

Definition 4.3. For a given property $\pi : \mathbb{D} \times \Theta \to \mathbb{R}^k$, we define the *$\pi$-distributional generalization (DG) gap* as follows:

$$\delta(\pi) \triangleq d_{\mathsf{TV}}(\pi\sharp P_1, \pi\sharp P_0) \tag{4.5}$$

This generic definition subsumes the version of DG defined in Definition 2.1 if we take $\pi(z;\theta) = (z;\theta)$, and extends classical on-average generalization in Section 1.2.

Evaluating distributional generalization enables us to assess the generalization of an ML model on the entire population, rather than on average. In Fig. 4.1 it is clear that the model's actual loss across the entire population is concentrated on a few individuals. Distributional generalization enables us to capture this discrepancy, whereas standard generalization does not.

As shown in Chapter 1, the ability of any classifier to successfully distinguish between observations of two classes can be characterized by the total variation between the class-conditional distributions of observations. By applying this fact to the worst-case MIA attackers, we can characterize vulnerability in terms of distributional generalization:

Proposition 4.3.1. The worst-case vulnerability to MIAs with adversary's features $\pi(z;\theta)$ is equal to the DG gap:

$$V(\mathcal{A}_\pi^*) = \delta(\pi).$$

According to Proposition 4.3.1, when the property function $\pi(z;\theta)$ matches the adversary's feature extraction mechanism, the DG gap is equal to the worst-case vulnerability to adversaries that use features $\pi(z;\theta)$.

Proof. The Bayes error R^* in the case of $\mathtt{Att}^*$ is:

$$R^* \triangleq \Pr[\mathtt{Att}^*(w) \neq m]$$

Recall that vulnerability is defined through the success probability of an adversary:

$$V(\mathcal{A}_\pi) \triangleq 2\Pr[\mathtt{Att}(w) = m] - 1$$

Thus, for a Bayes adversary, $V(\mathcal{A}_W^*)$ uses the complement of the Bayes error R^*:

$$V(\mathcal{A}_\pi) = 2(1 - \Pr[\mathtt{Att}^*(w) \neq m]) - 1 = 1 - 2R^*.$$

As outlined in Chapter 1, the Bayes error of the binary classifier under uniform prior is characterized by the TV distance between two class-conditional distributions. In our case, the class is the membership label m, thus:

$$\begin{aligned} R^* &= {}^1\!/\!_2 - {}^1\!/\!_2 \cdot d_{\mathsf{TV}}\left(\Pr_{\substack{S\sim P^n \\ z\sim S}}[\pi(z;T(S))],\ \Pr_{\substack{S\sim P^n \\ z\sim P}}[\pi(z;T(S))]\right) \\ &= {}^1\!/\!_2 - {}^1\!/\!_2 \cdot d_{\mathsf{TV}}\left(\pi\sharp P_1,\ \pi\sharp P_0\right), \end{aligned}$$

This implies the sought form.

□

This form is a straightforward consequence of our Bayes-optimal approach to vulnerability and is an application of a well-known result in statistical theory. It provides us with an intuitive interpretation of the worst-case vulnerability to MIAs—as it is equal to the distributional-generalization gap—thus with a guideline on how to prevent MIAs. The result holds for both white-box and black-box adversary models.

Let us visually illustrate distributional generalization and worst-case vulnerability. Consider adversary's features of the confidence scores $\pi((x,y);\theta) = h_\theta(x) \in [0,1]$. As

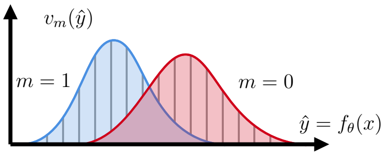


Figure 4.2: Distributional-generalization gap for models confidence scores $\hat{y} = h_\theta(x) \in [0, 1]$. The curves represent the probability density functions of models' outputs on the training datasets ($m = 1$) and outside ($m = 0$). The striped area shows the distributional-generalization gap: total variation between distributions of model's outputs on training and outside. Proposition 4.3.1 shows that the the size of the striped area exactly equals to the worst-case vulnerability to any adversary that uses model outputs $\hat{y}$ as features for distinguishing members from non-members.

the property function is continuous, the DG gap becomes (see Section 1.1):

$$\begin{aligned}\delta(\pi) &= d_{\mathsf{TV}}\left(\pi\sharp P_1, \pi\sharp P_0\right) \\ &= \frac{1}{2}\int_0^1 \left|v_1(\hat{y}) - v_0(\hat{y})\right| \mathrm{d}\hat{y},\end{aligned}$$

where v_1 and v_0 are probability density functions associated with probability distributions $\pi\sharp P_1$ and $\pi\sharp P_0$, respectively. See Fig. 4.2 for a visualization. The worst-case vulnerability to adversaries using features $\hat{y} = h_\theta(x)$ is the area between the densities of the "in" and "out" output distributions.

Note that the distance used in Proposition 4.3.1 is *average-dataset*. That is, when computing the features $\pi(z, \theta)$, the model θ is a random variable over the randomness of $T(\cdot)$ and $S \sim P^n$. To train models with minimal vulnerability to MIAs, Li et al. [113] used a similar yet different notion of distance, the distance between outputs of a *fixed* model on its training dataset and a validation dataset. Although conceptually similar, such distance cannot be directly used to evaluate the worst-case vulnerability using Proposition 4.3.1.

Overfitting and worst-case vulnerability. The absence of overfitting in the standard sense does not necessarily preclude MIAs [119, 194]. But, a straightforward implication of Proposition 4.3.1 shows there is a case when the standard generalization gap does bound the worst-case vulnerability:

Corollary 4.3.1. Let $\ell((x, y); \theta) = \mathbb{1}[f_\theta(x) \neq y]$ be the 0-1 loss, and the adversary's features be the loss values $\pi(z; \theta) = \ell(z; \theta)$. Then, the standard on-average generalization gap (see Section 1.2) equals worst-case vulnerability:

$$V(\mathcal{A}_\ell^*) = \left| \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim S}} \ell(z; T(S)) - \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim P}} \ell(z; T(S) \right| \tag{4.6}$$

Proof. As 0-1 loss is binary-valued, $\delta(\ell)$ simplifies by a property of the TV distance (see Section 1.1):

$$\begin{aligned}\delta(\ell) &= \left| \mathop{\mathbb{E}}_{L\sim\ell\sharp P_1}[L] - \mathop{\mathbb{E}}_{L\sim\ell\sharp P_0}[L] \right| \\ &= \left| \mathop{\mathbb{E}}_{(z,\theta)\sim P_1}[\ell(z;\theta)] - \mathop{\mathbb{E}}_{(z,\theta)\sim P_0}[\ell(z;\theta)] \right| \\ &= | \mathop{\mathbb{E}}_{\substack{S\sim P^n \\ z\sim S}}[\ell(z;T(S)] - \mathop{\mathbb{E}}_{\substack{S\sim P^n \\ z\sim P}}[\ell(z;T(S)]|,\end{aligned}$$

where the last transition is by definition of P_1 and P_0. □

Therefore, if a MIA adversary only observes whether a queried example has a correct or incorrect prediction by the target model, the upper bound on the success of any such attack has a direct relationship to standard overfitting. Thus, for such an adversarial model, no overfitting *does* imply no vulnerability to MIAs.

4.3.1 Disparate Vulnerability

In this section, we provide a theoretical analysis of vulnerability to MIAs disaggregated by subgroups.

We introduce a subgroup-specific version of distributional generalization, in which the distributions of the property π are computed on examples that belong to a given subgroup. We define the group-specific distributions:

$$\begin{aligned}P_{1,G} &\triangleq (z;T(S)), \text{ where } S\sim P^n, z\sim S_G, \\ P_{0,G} &\triangleq (z;T(S)), \text{ where } S\sim P^n, z\sim P_G.\end{aligned} \tag{4.7}$$

Definition 4.4. For a property function $\pi : \mathbb{D}\times\Theta\to\mathbb{R}^k$, the *DG gap* of group $G\in\mathbb{G}$ is defined as:

$$\delta_G(\pi) \triangleq d_{\mathsf{TV}}\Big(\pi\sharp P_{1,G},\ \pi\sharp P_{0,G}\Big),$$

Subgroup vulnerability from distributional generalization. To extend the worst-case analysis to subgroups, we use the worst-case subgroup vulnerability under adversary's features $\pi(z;\theta)$ to the corresponding Bayes adversary: $V_G(\mathcal{A}^*_\pi)$. We show that this subgroup vulnerability is also related to distributional generalization:

Proposition 4.3.2. The worst-case vulnerability of a subgroup G is bounded:

$$V_G(\mathcal{A}^*_\pi) \leq \delta_G(\pi) \tag{4.8}$$

Moreover, for subgroup-aware adversaries the bound becomes an equality:

$$V_G(\mathcal{A}^*_{\pi \,\circ\, g}) = \delta_G(\pi) \tag{4.9}$$

We defer the proof to Appendix A.3.

Formalizing disparate vulnerability. Finally, having discussed subgroup vulnerability, we can analyze disparate vulnerability. We define *disparity in vulnerability*:

Definition 4.5. Disparity in vulnerability (or disparity for short) between two subgroups G and G' is the difference in vulnerability of these subgroups:

$$\Delta V_{G,G'}(\mathcal{A}^*_\pi) \triangleq V_G(\mathcal{A}^*_\pi) - V^*_{G'}(\mathcal{A}^*_\pi)\,.$$

The previous results on the connection between subgroup vulnerability and distributional generalization enable us to relate disparity to degrees of distributional generalization across different population subgroups. From Proposition 4.3.2, we can see that the magnitude of disparity can be trivially bounded using distributional-generalization gaps of the involved subgroups:

Corollary 4.3.2. Magnitude of disparity between subgroup G and G' is upper bounded:

$$\left|\Delta V_{G,G'}(\mathcal{A}^*_\pi)\right| \leq \max\{\delta_G(\pi), \delta_{G'}(\pi)\} \tag{4.10}$$

Moreover, disparity has an exact closed form for subgroup-aware adversaries:

Corollary 4.3.3. Suppose that a subgroup-aware adversary uses features (π, G). Then, disparity between subgroups G and G' is the difference between distributional generalization gaps of these subgroups:

$$\Delta V_{G,G'}(\mathcal{A}^*_{\pi \,\circ\, g}) = \delta_G(\pi) - \delta_{G'}(\pi)\,. \tag{4.11}$$

4.3.2 Takeaways

Necessary and sufficient condition for MIA vulnerability. Without making any parametric assumptions, we have showed that the vulnerability to MIAs can be characterized using an extended notion of generalization, and that disparity is bounded by the difference in levels of distributional generalization across population subgroups. This interpretation of a standard result in statistical theory generalizes and complements the theoretical findings of Yeom et al. [194] and Sablayrolles et al. [150]. It also confirms that the presence of standard overfitting is not a necessary condition for MIAs to succeed [119, 194].

Hardness of defending against MIAs. The interpretation of worst-case vulnerability through distributional generalization has important consequences for practical defences against MIA that do not rely on differential privacy.

In order to reduce the vulnerability against adversaries that use features $w = \pi(z; \theta)$, the distribution of w for examples that are outside of the training set has to be close to that for the training set examples. This means that, to avoid vulnerability, a target model has to—either implicitly or explicitly—learn the distribution of w [94] which is a stronger requirement than what is typically necessary for its main task (i.e. generalization in terms of accuracy, or average error).

Moreover, adversaries are not limited to one set of features; thus, the distribution has to be learned for a multitude of possible configurations of adversary's features. Additionally, to prevent disparity in vulnerability, the distribution of w has to be learned across population subgroups—an even more challenging task.

4.4 Detecting and Measuring Disparate Vulnerability

We showed in Section 4.3 that vulnerability to MIAs appears when a model lacks in distributional generalization. The degree to which records are vulnerable can vary across subgroups in the data, potentially resulting in disparate vulnerability. In this section, we provide mechanisms to reliably estimate subgroup vulnerability and its disparity in practice.

To empirically estimate MIA vulnerability, we simulate the MIA game with a real attack. If we could play the game infinite times, then estimating the success probability of the adversary would be trivial. In practice, however, we can only run the game a finite amount of times, which provides us with a finite number of challenge examples z. We group these examples into two sets of datasets of n elements: a set of r datasets $\{S_i\}_{i=1..r}$ composed of n "in" examples (i.e., sampled as in line 4 of the MIA game, used for training), and r datasets $\{\bar{S}_i\}_{i=1..r}$ composed of n "out" examples (i.e., sampled as in line 6 of the MIA game, not used for training). Each pair of datasets S_i and $\bar{S}_i$ can be seen as the train and test datasets of one model.

We define the estimate of vulnerability as:

$$\hat{V}(\mathcal{A}) \triangleq \frac{1}{r} \sum_{i=1}^{r} v_i \tag{4.12}$$

where v_i is the *model-specific estimate of vulnerability*: the advantage of the adversary against a single target model. We compute v_i for a pair of datasets S_i and $\bar{S}_i$ as:

$$v_i \triangleq 2 \cdot \frac{1}{2n} \left(\sum_{j=1}^{n} \mathbb{1}[\mathcal{A}(S_i^{(j)}, T(S_i), n, P) = 1] + \sum_{j=1}^{n} \mathbb{1}[\mathcal{A}(\bar{S}_i^{(j)}, T(S_i), n, P) = 0] \right) - 1, \tag{4.13}$$

As r increases, $\hat{V}(\mathcal{A})$ approximates the value of the true vulnerability V.

We use the same approach to estimate subgroup vulnerability $V_G(\mathcal{A})$, but we only use examples that belong to the subgroup of interest G when computing the model-specific estimate of subgroup vulnerability $v_{i,G}$. We omit $\mathcal{A}$ when it is clear from context.

4.4.1 Statistical Detection of Disparity

When evaluating subgroup vulnerability, we have to rely on subsets of $(S_i, \bar{S}_i)$ formed by subgroup examples. These subsets are possibly of size much smaller than n. Due to the variance of the empirical averages in the Eq. (4.13), an estimate of subgroup vulnerability is in general less statistically reliable than the estimate of overall vulnerability that uses datasets $(S_i, \bar{S}_i)$ in their entirety. As a result, when estimating disparate vulnerability using the estimates of subgroup vulnerability, we need to statistically ensure that, if found, disparity is not due to random chance.

More formally, given estimates $\{v_{i,G}\}_{i=1..r}$ across different subgroups, we want to find statistical evidence that the actual subgroup vulnerabilities differ:

$$V_{G_1} \overset{?}{\neq} V_{G_2} \overset{?}{\neq} \ldots \overset{?}{\neq} V_{G_t} \tag{4.14}$$

Multiple subgroups. This problem is an instance of a standard within-subjects experimental design: We have multiple measurements (model-specific vulnerability estimates for different subgroups $v_{i,G_1}, v_{i,G_2}, \ldots, v_{i,G_t}$) for the same subject (model $T(S_i)$). We want to know whether the means of vulnerability values differ across subgroups. Therefore, we can determine whether the training algorithm exhibits disparate vulnerability using the repeated-measures one-way anova model (see, e.g., Seltman [156, Chapter 14]). This approach enables us to use the anova F-test to establish whether there is evidence of disparate vulnerability. Following the standard protocol, if the F-test is positive, we perform *post-hoc followup tests* to determine which particular pairs of subgroups exhibit disparity. For the post-hoc tests, we use pairwise dependent t-tests with correction for multiple comparisons. As the correction method, we use the standard Benjamini-Hochberg procedure for controlling the false detection rate.

Two subgroups. When comparing only two subgroups, G and G', the procedure naturally simplifies to running one dependent t-test that checks if the difference between means of two groups is significant.

4.4.2 The Bias Problem

Some attacks in the literature assume that the adversary has *additional knowledge* beyond the tuple $(z; \theta, T, n, P)$. This knowledge can result in the vulnerability estimation being positively biased: indicating higher vulnerability than the actual worst case within the knowledge model of $(z; \theta, T, n, P)$. Overestimating vulnerability is not necessarily an issue, as pessimistic estimates incentivize caution in deployment. However, if the positive bias is correlated with the parameters of a subgroup (e.g., higher bias for smaller subgroups), it leads to incorrect conclusions about *disparate* vulnerability.

In this section, we check whether estimates of vulnerability using attacks proposed in the literature are biased. We evaluate three attacks:

- **Shadow-model attack [161].** An adversary trains a number of shadow models using the target training algorithm $T(\cdot)$ on datasets sampled from P^n. The adversary uses these shadow models to train a machine-learning classifier to guess the membership from observed features. In our evaluation, we use 30 shadows and Gradient Boosting Trees as the attack classifier.
- **Average-threshold attack [194].** An adversary has additional knowledge: the average loss on the training dataset τ and the loss function $\ell(z; \theta)$ used to compute this average, $(\tau,\ \ell(z; \theta))$, where $\tau \triangleq \frac{1}{n} \sum_{z \in S} \ell(z; \theta)$. When attacking, the adversary uses τ as threshold to decide whether the challenge example was "in" (the example's loss less than threshold) or "out" (greater than threshold).
- **Optimal-threshold attack [31, 162].** An adversary has additional knowledge: the loss function ℓ and the optimal loss threshold τ^* that separates the losses in the best way, $(\tau^*,\ \ell(z, \theta))$, where

  $$\tau^* \triangleq \arg\max_{\tau} \frac{1}{n} \sum_{z \in S} \mathbb{1}[\ell(z; \theta) \leq \tau] + \mathop{\mathbb{E}}_{z \sim P} \left[\mathbb{1}[\ell(z; \theta) > \tau]\right]$$

  The attack proceeds as the average-threshold one.

We deviate slightly from the attacks' original formulations. The threshold attacks use $\pi(z; \theta) = \ell(z; \theta)$ as features, where the loss function is cross-entropy, whereas the original shadow-model attack used $\pi((x, y); \theta) = (f_\theta(x), y)$. For fairness of the comparison, we make all adversaries use the threshold attacks' features.

As we want to evaluate subgroup-aware adversaries, we use features $\pi(z; \theta) = (\ell(z; \theta), g(z))$ for all attacks, with cross-entropy as loss function. We make the attacks subgroup-aware as follows. For the shadow-model attack, the adversary trains separate attack classifiers for each subgroup, and then applies the appropriate classifier to each challenge example. For the threshold attacks, we assume the adversary has different

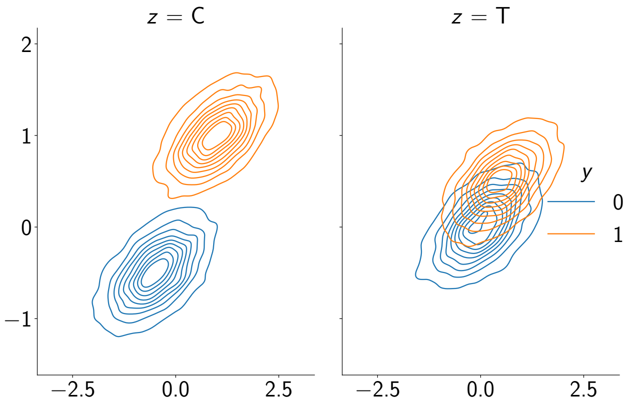


Figure 4.3: Distribution of values in our synthetic data. *x-axis:* value of the 1-st dimension of the synthetic data, *y-axis:* value of the 2-nd dimension. We use 100-dimensional data for our experiments.

thresholds for each subgroup [31, 163], i.e., average loss, respectively optimal threshold, per subgroup.

Method. It is hard to tell exactly if an estimate is higher than the worst-case vulnerability, as in practice the worst case is unknowable. We propose a simple test for bias within our adversarial model: run the estimation method against *data-independent models*. A target model can be independent of its training data, e.g., if it is completely random, constant, or trained with differential privacy parameter $\epsilon \approx 0$ (see Section 4.5.2). If the model is independent of the data, we expect the estimates of overall and subgroup vulnerabilities, as well as disparity, to all be zero in expectation. We refer to any violation of this property as *null-model bias*. We are not only interested in whether a method exhibits such bias, but in whether this bias is correlated with subgroups.

Dataset. To have control over the distributions of subgroups and their representation, we create a synthetic dataset. We assume that the examples have binary class labels $y \in \{0, 1\}$, and belong to one of two subgroups $G \in \{\texttt{ctrl}, \texttt{treatment}\}$. We generate the examples from the multivariate normal distributions:

$$\begin{aligned}
\Pr(z \mid y = 0, G = \texttt{ctrl}) &\sim \mathcal{N}(-{}^1\!/\!_2 \cdot \mathbf{1}^d, \Sigma) \\
\Pr(z \mid y = 1, G = \texttt{ctrl}) &\sim \mathcal{N}(1 \cdot \mathbf{1}^d, \Sigma) \\
\Pr(z \mid y = 0, G = \texttt{treatment}) &\sim \mathcal{N}(0 \cdot \mathbf{1}^d, \Sigma) \\
\Pr(z \mid y = 1, G = \texttt{treatment}) &\sim \mathcal{N}({}^1\!/\!_2 \cdot \mathbf{1}^d, \Sigma),
\end{aligned}$$

where $\mathbf{1}^d$ is a d-dimensional vector of all ones, and the covariance matrix Σ is generated such that $||\Sigma||_{\max} \leq 1$. We use $d = 100$ dimensions, and set $\Pr[y = 1] = {}^1\!/\!_2$. See Fig. 4.3 for an illustration.

To reflect that some subgroups can be harder to learn than others, the distributions are designed in such a way that the subgroup $G = \texttt{ctrl}$ is more separable and hence more easily learnable than the subgroup $G = \texttt{treatment}$. In our experiments we use the subgroup $G = \texttt{ctrl}$ as the control (or *majority*) subgroup with fixed number of representatives in the data, and $G = \texttt{treatment}$ as the treatment (or *minority*) subgroup whose size we vary.

Setup. To see if the potential null-model bias depends on the sizes of subgroups, we generate multiple synthetic datasets such that each contains data belonging to two subgroups: control and treatment. The control subgroup has 1000 representatives in each dataset; the size of the treatment subgroup varies between 25 and 1000, with 8 distinct values. We run 8 experiments with different subgroup proportions. Within each experiment, we train 200 target models on freshly generated datasets. We set the target training algorithm to output the same classifier for any input training dataset. Recall that because the models are independent of the input, we expect all vulnerability estimates to be zero on average. We estimate disparity using three attacks described above, and run t-tests to see if the estimates are statistically significant as explained in Section 4.4.1.

Results on our synthetic dataset. In Fig. 4.4, we can see that the estimates of disparity produced with the shadow-model attack and the average-threshold attack are centered around zero, with the statistical tests confirming no significant difference from zero. The estimates coming from the optimal-threshold attack, however, are highly biased compared to the other attacks, as the estimates are consistently and significantly ($p < 0.001$) different from zero. The bias is always positive — overestimates disparity — and gets higher as the size of the treatment subgroup decreases. As the target models are independent of their training data and thus cannot have disparate vulnerability, we conclude that the use of the optimal-threshold attack results in significant null-model bias that grows as the subgroup size gets smaller.

Results on the dataset by Chang and Shokri [31]. To verify that our results are not artifacts of our specific synthetic data setup, we also reproduce the data setup used by Chang and Shokri to evaluate their subgroup-aware optimal-threshold attack. In their setup, they have one fixed dataset containing four subgroups that we denote as “0-0”, “0-1”, “1-0”, “1-1”, where the first number indicates simulated demographic group and the second number the class y (we refer to the original work [31] for details). The subgroups have 50, 450, 1000, and 1000 examples, respectively, with the total dataset size of 2500 examples. Following Chang and Shokri, we randomly subsample training datasets of size 1250 from the full dataset, and train one model on each. As before, we “train” a data-independent model. In this experiment, we only use threshold attacks due to the small size of the dataset (see Section 4.6 for more details). We use the anova F-test as described in Section 4.4.1 to determine whether any of the subgroups have differing subgroup vulnerabilities.

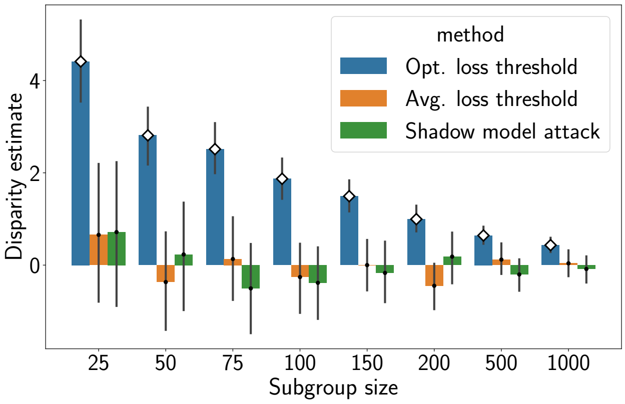


Figure 4.4: Null-model bias of methods to estimate disparate vulnerability. Disparity in percentage points (*y-axis*) vs. size of the treatment subgroup in the training data (*x-axis*). Computed on synthetic datasets with fixed control subgroup (1000 examples) .The target training algorithm is data-independent: actual MIA vulnerability, subgroup vulnerabilities, and disparity in vulnerability are all zero. The error bars represent the variation across 200 model-specific estimates. The diamond marker (◊) means that an estimate significantly differs from zero with $p < 0.001$.

Fig. 4.5 shows that significant null-model bias of the optimal-threshold attack also appears on this dataset (F-test $p < 0.001$). In particular, the subgroup vulnerability for the smallest subgroup "0-0" with 50 examples appears as 4%. At the same time, the estimates from the average-threshold attack are centered around 0 and do not significantly differ (F-test $p \approx 0.1$), suggesting no null-model bias.

This bias, however, should not affect the conclusions by Chang and Shokri [31]. Rather than directly using the estimates of subgroup vulnerability, their analysis used *differences* in estimates of subgroup vulnerability between two models (a "fair" and a "regular" model). In their particular scenario, the bias introduced by the estimation should be cancelled out in the final difference. Although the conclusions of Chang and Shokri should not be affected by the bias, estimation methods such as the optimal-threshold attack should be avoided when evaluating disparate vulnerability in general.

Takeaways. Biased estimators of vulnerability can result in consistent overestimation of disparity if the bias correlates with subgroup parameters. The shadow-model attack does not have such bias as it does not have access to any information about a specific target. Interestingly, the average-threshold attack, despite using an additional piece of knowledge that goes beyond our adversarial model, also does not exhibit such bias. On the contrary, the optimal-threshold attack produces significantly biased estimates for small groups.

Our results show the need to evaluate bias of the estimation method when measuring

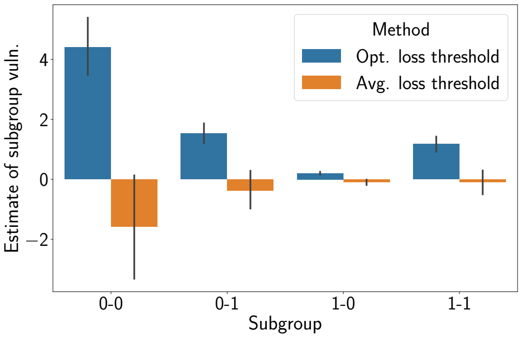


Figure 4.5: Null-model bias on the synthetic data setup from Chang and Shokri [31]. Estimate of disparity in percentage points (*y-axis*) vs. subgroup (*x-axis*). The target training algorithm is data-independent, thus actual MIA vulnerability, subgroup vulnerabilities, and disparity in vulnerability are all zero.

disparate vulnerability. To this end, we proposed to measure null-model bias, which detects bias when the worst-case vulnerability is zero. This test does not preclude a method from having bias if the worst-case vulnerability is larger. However, in practice MIA vulnerability has been shown to be relatively low.

4.4.3 Does Disparate Vulnerability Exist in ML Models?

Having established suitable methods for measuring disparate vulnerability, we apply them in a synthetic setup, and show that disparate vulnerability arise in practice.

Setup. To capture the effect of subgroup size in the training data, we create several experiments with different subgroup proportions. Within each experiment, we sample 200 dataset pairs S_i and $\bar{S}_i$ from our data distribution. In each dataset, the size of the control subgroup is fixed at 2500, and we vary the size of the treatment subgroup between experiments: 100, 500, 1000, and 2500. We estimate subgroup vulnerabilities using the subgroup-aware shadow-model attack (see Section 4.4.2), because this attack is guaranteed to not have null-model bias. As before, we use $\pi(z;\theta) = (\ell(z;\theta), g(z))$ as adversary's features. To train shadow models, we independently sample 30 fresh datasets from our data distribution. We use t-tests to determine whether measured disparity is statistically significant (see Section 4.4.1).

Targets. We evaluate the following model families: logistic regression, and two ReLU neural networks with one hidden layer containing 8 and 32 neurons, respectively. We use the *scikit-learn* library [140] to train these models. All our models attain close to 100% test accuracy in our synthetic data setup.

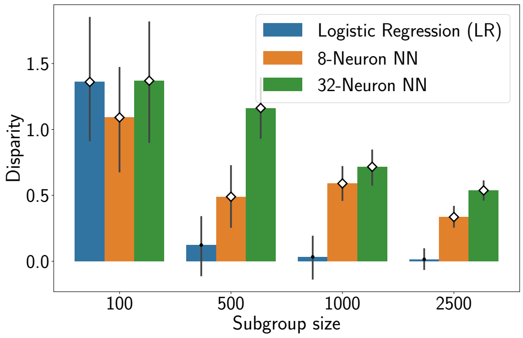


Figure 4.6: Disparate vulnerability vs. subgroup representation in a training dataset. The *y-axis* represents disparity in vulnerability between the treatment group G and control group G' whose size is fixed to 2500, in percentage points. The error bars represent the variation across 200 model-specific estimates. Statistical significance markers: $p < 0.001$ ($\Diamond$), $p < 0.01$ ($\circ$), $p \geq 0.01$ ($\cdot$).

Results. The results in Fig. 4.6 show that ML models can exhibit disparate vulnerability, even on a simple dataset. For all treatment sizes and targets, our estimates of disparity are significant ($p < 0.001$), with the exception of the logistic regression when the treatment subgroup is relatively well-represented (500 – 2500 examples). We also see that the sample size of the subgroup plays an important role in disparate vulnerability: *the less represented is a group in the training data, the higher the disparate vulnerability as compared to a better represented group.* Even though the sample size seems to be the dominant effect, we observe small but significant disparate vulnerability even when the subgroups are equally represented in training.

4.5 Mitigating Disparate Vulnerability

We study whether existing methods for addressing privacy and fairness in ML prevent disparate vulnerability.

4.5.1 Fairness Constraints

Due to the dependency of disparate vulnerability on the disparate *behavior* of the model across subgroups, minimizing the between-subgroup discrepancy in any given property, such as model's outputs or loss [38], intuitively could decrease disparate vulnerability.

Formally, let us denote by $\delta_{G,G'}(\pi)$ the total-variation distance between distributions of

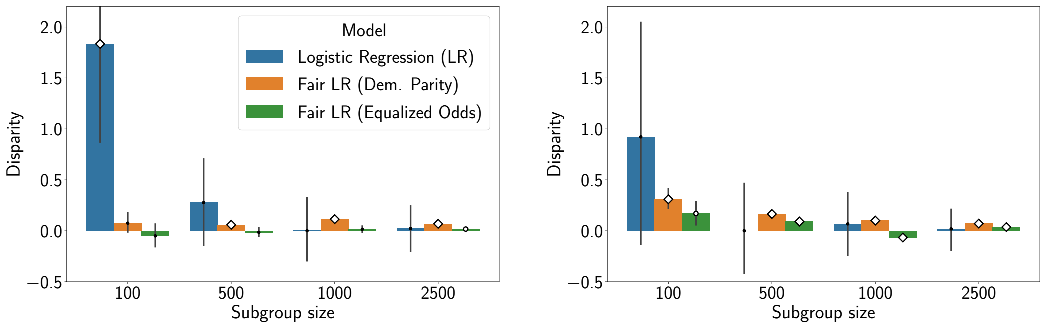


Figure 4.7: Effect of algorithmic-fairness constraints on disparate vulnerability. The vulnerability is estimated with subgroup-aware attacks that use models' outputs as the feature *(left)*, and the models' loss *(right)*. The results for logistic regression are provided for reference (its values here are not comparable with the results of other experiments as the data dimensionality is different). See Fig. 4.6 caption for details.

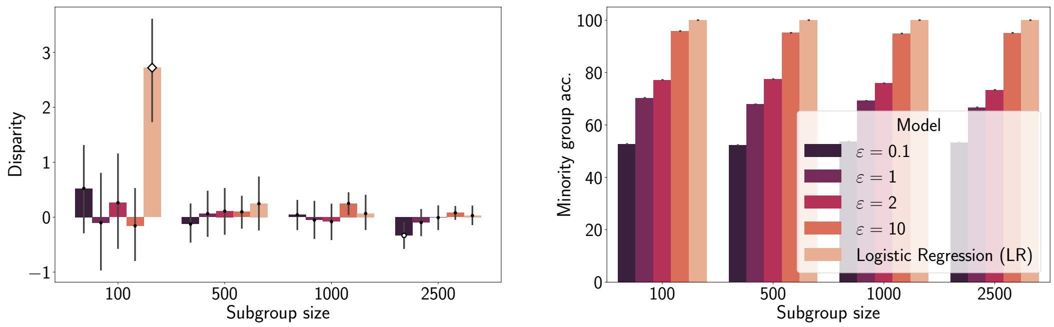


Figure 4.8: Effect of differentially private training on disparate vulnerability *(left)*, and test accuracy *(right)*. The results for logistic regression are provided for reference. See Fig. 4.6 caption for details.

some property of a model $\pi(z;\theta)$ on examples coming from two subgroups G and G':

$$\delta_{G,G'}(\pi) \triangleq d_{\mathsf{TV}}\left(\Pr_{\substack{S\sim P^n\\ z\sim P}}[\pi(z;\theta) \mid z \in G], \Pr_{\substack{S\sim P^n\\ z\sim P}}[\pi(z;\theta) \mid z \in G']\right).$$

Certain notions of algorithmic fairness provide an upper bound, or are equivalent to, the above gap given an appropriate choice of the property function: if we choose the model property to be its outputs, then with $\pi((x,y);\theta) = f_\theta(x)$, we obtain *demographic parity* [55]. Similarly, for the 0-1 loss property of the model, choosing $\pi((x,y);\theta) = \mathbb{1}[f_\theta(x) \neq y]$ gives us *accuracy equality*.

In practice, a notion of fairness is satisfied on the training dataset rather than the whole data distribution. To capture this, we define an in-training gap as follows:

$$\hat{\delta}_{G,G'}(\pi) \triangleq d_{\mathsf{TV}}\left(\Pr_{\substack{S\sim P^n \\ z\sim S}}[\pi(z;\theta) \mid z \in G], \Pr_{\substack{S\sim P^n \\ z\sim S}}[\pi(z;\theta) \mid z \in G']\right).$$

The following proposition establishes that, if the in-training gap is bounded and the model generalizes its fairness condition well, then vulnerability disparity is bounded to adversaries that use the property addressed by the fairness notion:

Proposition 4.5.1. Suppose a subgroup-aware adversary uses features $\pi \circ g$, and the following two conditions are satisfied:

1. Fairness on the training data: $\hat{\delta}_{G,G'}(\pi) \leq \eta$
2. On-average fairness generalization: $|\delta_{G,G'}(\pi) - \hat{\delta}_{G,G'}(\pi)| \leq \nu$

Then, the magnitude of disparity in worst-case vulnerability is bounded as follows:

$$|\Delta V_{G,G'}(\mathcal{A}^*_{\pi,g})| \leq 2\eta + \nu.$$

We defer the proof to Appendix A.3.

We note that these guarantees only apply to adversaries targeting the features addressed by the implemented the fairness notion. In other words, just as in algorithmic-fairness literature where no single fairness measure is appropriate in a general context [65], no one fairness measure can provide guarantees for bounding disparate vulnerability for any adversary.

Empirical evaluation

Fairness notions. To validate the theoretical results, we estimate vulnerability of models that satisfy two algorithmic-fairness notions: First, *demographic parity* [55] which ensures that distributions of model outputs between demographic subgroups are close: $\delta_{G,G'}(\hat{y}) \approx 0$. Second, *equalized odds*, which ensures that true-positive rates and false-positive rates between the subgroups are close [78]. We choose these notions as they are common in the literature, and there exist efficient algorithms and tooling for producing classifiers that satisfy them. To train the classifiers, we use the threshold post-processing approach [78], applied to a logistic regression classifier.

Setup. Within the setup of Section 4.4.3, we run the following two experiments:

E1 We fulfill the requirements of Proposition 4.5.1. For this, we estimate vulnerability using features equalized by demographic parity: $\pi((x, y); \theta) = (f_\theta(x), g(z))$. By

Proposition 4.5.1, we expect low disparity in vulnerability *for both classifiers* as long as they generalize their fairness property well. In Appendix A.3, we show that in our data setup equalized odds implies demographic parity, thus the theoretical guarantee also applies for equality of odds.

E2 We estimate vulnerability using adversary's features $\pi(z; \theta) = (\ell(z; \theta), g(z))$ which do *not* match what the fairness property does, so the requirements of Proposition 4.5.1 are not fulfilled.

We find that with 100 dimensions in our setup, the threshold-optimization algorithm produces models that classify the data with 100% accuracy and no vulnerability. To demonstrate a setting where disparate vulnerability arises, we deviate from the parameters of Section 4.4.3 and use the synthetic dataset with 10 dimensions.

Results. We present the results in Fig. 4.7. For E1, we see that demographic parity decreases disparate vulnerability compared to standard logistic regression. This empirically confirms Proposition 4.5.1. For E2, as expected, both equalized odds and demographic parity do not completely prevent disparate vulnerability. Yet, they do decrease its magnitude by $3\times$ compared to the standard logistic regression.

In our particular setup, the constrained models do not perform worse than the unconstrained models. In general, however, fairness notions can be inherently at odds with accuracy [203].

4.5.2 Differentially Private Training

In this section, we look at how learning with differential privacy (see Chapter 1) relates to disparity in vulnerability.

DP training limits the contribution of any individual in the dataset to the model training. Thus, DP should decrease vulnerability to MIAs. In particular, Yeom et al. [194] and Humphries et al. [87], showed the advantage of a MIA adversary is bounded by DP in the setting of the MIA game. For example:

Proposition 4.5.2 (Adapted from Yeom et al. [194])**.** If the training algorithm satisfies ϵ-DP, the worst-case vulnerability with any adversary's features W is bounded:

$$V(\mathcal{A}^*_W) \leq \exp(\epsilon) - 1 \tag{4.15}$$

These guarantees extend to disparate vulnerability under a technical condition that aims to avoid undefined behavior:

Proposition 4.5.3. For any two given subgroups $G, G' \in \mathbb{G}$, suppose that the dataset sampling in the MIA game (see Section 4.2) ensures that each S has some representatives of each subgroup. Formally, we condition the sampling on $|S_G| > 0$ and $|S_{G'}| > 0$. Moreover, if the training algorithm satisfies ϵ-DP, then the worst-case subgroup vulnerability of G, as well as magnitude of vulnerability disparity between G and G', is uniformly bounded for any adversary's features $\pi(z; \theta)$:

$$V_G(\mathcal{A}^*_\pi) \leq \frac{\exp(\epsilon) - 1}{\exp(\epsilon) + 1}, \quad \left|\Delta V_{G,G'}(\mathcal{A}^*_\pi)\right| \leq \frac{\exp(\epsilon) - 1}{\exp(\epsilon) + 1}. \tag{4.16}$$

We defer the proof to Appendix A.3.

Empirical evaluation. To study how DP affects disparate vulnerability we train DP models with different privacy levels. As target models, we use DP logistic regression trained using the objective perturbation method [33]. We use a min-max scaler, and provide a maximum row norm estimated on a separate sample from the data distribution. We use privacy levels $\epsilon = 0.1, 1, 2, 10$.

We see in Fig. 4.8 that, for all evaluated values of ϵ, DP training considerably reduces disparity compared to the non-private logistic regression, with statistical tests not detecting significant deviations from 0.

On the downside, unlike training with fairness constraints, DP training results in a significant decrease in accuracy of the models: from 45 p.p. to 5 p.p. drop depending on the value of ϵ.

4.5.3 Takeaways

Fairness only bounds disparate vulnerability in certain scenarios. Even when the classifier's fairness property generalizes beyond the training set, the bound is restricted to the adversarial strategy covered by the chosen fairness notion. Covering one adversarial strategy, however, is a weak security guarantee: the model could be (disparately) vulnerable to other strategies. Moreover, it is known that different fairness constraints are at odds with each other [65]. Hence, a model protected by one fairness notion may be inherently insecure against adversaries exploiting non-protected features.

Differential privacy bounds disparate vulnerability. We show that DP provides an upper bound on the vulnerability of all individuals, subgroups, and therefore on disparate vulnerability too. On the flip side, because DP guarantees are often at odds with accuracy, in practical applications ϵ is usually set high, allowing for a lot of variation within the upper bound of Proposition 4.5.3. Practically, the particular approach to

DP training that we evaluated has mitigated disparity even with a high privacy level $\epsilon = 10$ that results in vacuous theoretical bounds, but at significant accuracy costs.

4.6 Evaluation on Real-World Data

To investigate if we can detect disparate vulnerability in a realistic setting, we use the following two datasets as case studies:

- *ADULT dataset* [100]. The dataset contains 48,842 examples from the 1994 Census database[1]. The task is to determine if a yearly salary is over/under $50K. It contains attributes such as age, sex, education, race, native country, etc. After one-hot encoding, the dataset contains 91 features. We use the race column as the subgroup attribute.
- *Texas-50K dataset.* We create this dataset based on 2013 Texas Hospital Discharge data[2]. As our evaluation setup is computationally expensive, to accommodate the same training algorithms as used in the synthetic data experiments, we randomly subsample 50,000 examples, and reduce the number of features for training. We use the following columns: type of admission, illness severity, mortality risk, principal diagnosis code (out of more than 6000 codes, we only keep the top 1000 and create one separate code for the rest), length of stay, and patient's demographic attributes: sex, race, ethnicity. After one-hot encoding, we have 1025 features. We use the race column as the subgroup attribute. As a task, analogously to the ADULT dataset, we use prediction of whether the total amount of charges is greater than a threshold (e.g., for health-insurance risk-scoring). As the threshold we pick the median total charges on the subsampled dataset.

Table 4.1 provides details about the subgroups.

Target models. We consider as target models logistic regression and neural networks with 8 and 32 neurons in the hidden layer (Section 4.4.3), logistic regression with fairness constraints (Section 4.5.1), and differentially private logistic regression with ϵ values 1, 2, and 10 (Section 4.5.2). All the models beat the random accuracy baseline on the tasks.

Estimation method. As opposed to our synthetic data setup in which datasets to train shadow models can be directly sampled from the data-generating distribution, when real data is involved we can only sample data from the available finite dataset. We split

[1] https://archive.ics.uci.edu/ml/datasets/adult

[2] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm

Table 4.1: Subgroup representation in the datasets.

Dataset	G	Size
ADULT	"White" (WH)	38,903
	"Black" (BL)	4,228
	"Asian-Pac-Islander" (AI)	1,303
	"Amer-Indian-Eskimo" (AE)	435
	"Other" (OT)	353
	All	48,842
Texas-50K	4	31,514
	5	10,883
	3	6,451
	2	1,019
	1	133
	All	50,000

the dataset in two parts: one for training of the shadow models, and one for evaluation of vulnerability [161]. As a result, the amount of available training data is greatly reduced, in particular, for minority subgroups that already have few representatives in the dataset. To avoid this problem, in this section we use the average-threshold attack for vulnerability estimation, which does not require training shadow models. Our evaluation in Section 4.4.2 showed that this attack is not null-model biased.

Setup. To train each target model, we randomly subsample 50% of the dataset to use for training (S_i), and hold out the remaining data ($\bar{S_i}$). We train 200 models for each model family on different splits of the dataset. For our statistical tests (see Section 4.4.1), we use $\alpha = 0.01$ as significance level.

Results. We summarize the results in Table 4.2. As in our synthetic experiments, we observe evidence of disparity in neural networks. Importantly, the results show that low vulnerability in absolute terms does not imply absence of disparity. On ADULT, the 8-neuron network shows relatively low 0.4% vulnerability but statistically significant disparity ($p < 10^{-4}$). Interestingly, on Texas-50K, we also see statistical evidence of disparate vulnerability for logistic regression with demographic-parity constraints, although its overall vulnerability of 1.46% is comparable to standard logistic regression.

For the models with F-test $p < 0.01$, we conduct follow-up post-hoc tests to see which particular pairs of subgroups have high disparity (we defer the detailed results of the post-hoc tests to Appendix D.2). On ADULT, consistently with our synthetic experiments, the smaller subgroups "Asian-Pac-Islander" (AI, 1,302 examples), and "Other" (OT, 353 examples), exhibit disparity between themselves and other more populous subgroups. On Texas-50K, almost all subgroup pairs exhibit significant disparity for 32-neuron network.

Table 4.2: Summary of models performance and vulnerability on ADULT and Texas-50K. Columns: *Disparity test:* p-value of the anova F-test that checks if any of the subgroups have differing subgroup vulnerabilities, *Test acc.:* Test accuracy of models, *Gen. gap:* Per-model difference between train accuracy and test accuracy, *Vuln.:* Aggregate vulnerability $V(\mathcal{A})$. Bold font indicates models that have statistically significant disparity ($p < 0.01$).

ADULT	Disparity test	Test acc.		Gen. gap		Vuln., %	
	p	avg	std	avg	std	avg	std
Model							
Logistic Regression (LR)	0.3230	0.8404	0.0018	0.0012	0.0034	0.0942	0.4093
8-Neuron NN	**0.0000**	0.8421	0.0018	0.0044	0.0033	0.4052	0.3927
32-Neuron NN	**0.0000**	0.8410	0.0019	0.0131	0.0033	1.1373	0.4178
DP LR, $\epsilon = 1$	0.8534	0.7797	0.0135	0.0006	0.0040	0.0830	0.3478
DP LR, $\epsilon = 2$	0.0500	0.8053	0.0076	0.0004	0.0036	0.0563	0.3360
DP LR, $\epsilon = 10$	0.0419	0.8321	0.0023	0.0011	0.0032	0.0888	0.4100
Fair LR (Dem. Parity)	0.8945	0.8267	0.0018	0.0011	0.0035	0.0980	0.3331
Fair LR (Equalized Odds)	0.7089	0.7941	0.0095	0.0006	0.0038	0.0782	0.3521

Texas-50K	Disparity test	Test acc.		Gen. gap		Vuln., %	
	p	avg	std	avg	std	avg	std
Model							
Logistic Regression (LR)	0.2666	0.7833	0.0021	0.0152	0.0036	1.3905	0.4374
8-Neuron NN	0.0112	0.8836	0.0068	0.0282	0.0055	2.2384	0.5916
32-Neuron NN	**0.0000**	0.8639	0.0060	0.0686	0.0060	6.6238	0.7212
DP LR, $\epsilon = 1$	0.6192	0.6175	0.0191	0.0002	0.0045	0.0540	0.4317
DP LR, $\epsilon = 2$	0.0522	0.6363	0.0136	0.0014	0.0040	0.2125	0.3916
DP LR, $\epsilon = 10$	0.9737	0.7114	0.0146	0.0038	0.0041	0.5224	0.3245
Fair LR (Dem. Parity)	**0.0078**	0.7609	0.0028	0.0143	0.0039	1.2393	0.3444
Fair LR (Equalized Odds)	0.7174	0.7477	0.0180	0.0133	0.0038	1.4676	0.3983

The results for the logistic regression with fairness constraints are unlike the synthetic experiments. As opposed to a minority subgroup, as in the previous results, disparity appears between the most populous subgroup “4” (31,514 examples) and subgroups “2”, “3” and “5”. This disparity does not exist in the standard logistic regression. Thus, this result shows that fairness constraints can introduce disparity when the conditions of Proposition 4.5.1 are not met.

Discussion. We have used binary classification tasks for compatibility with the fairness definitions, but we expect disparity to be more pronounced in multi-class settings. As detailed in Section 4.3.2, disparate vulnerability is bound to happen whenever a model does not faithfully learn the distributional properties of the data for some subgroups. Prior research suggests it is likely to appear when the task has many features, or many classes in the case of classification [153].

We also only considered relatively small dataset sizes. Bigger datasets, on the one hand, enable better learning of the models thus decreasing vulnerability and disparate vulnerability, but on the other hand, they would enable the adversary to use shadow-model attacks that could provide better results than the average-threshold attack used in our experiments.

We leave investigations of the effect of the number of classes and dataset size on disparate vulnerability for future work.

4.7 Related Work

Theory studies on MIA. Yeom et al. studied the relation of MIAs to overfitting [194]; in their work, they formalize MIA as an indistinguishability game, which we adapt to construct our theoretical framework. Farokhi et al. analyzed the dependence of MIA's success on the amount of information the model memorizes [62], and Jayaraman et al. investigated their dependence on the prior probability that the example given to the adversary is a member or non-member of the training set [90]. Yeom et al. [194], and Cherubin et al. [36] showed that MIAs success is bounded by DP. Humphries et al. [87] showed these bounds only apply so long as the training data are i.i.d.-sampled. All these analyses, however, are only meaningful for the *average*-case MIA. A classifier thought to be secure according to these analyses may provide weaker protection to certain individuals or subpopulations. Our work complements these studies and generalizes the notion of MIA risk to *subgroups* of the population, enabling study of vulnerability for subsets of the records' labels, individuals, and subpopulations.

Disparate impact. The work on disparity in machine learning is centered on understanding and mitigating disparate impact of algorithmic decisions on subpopulations [11, 37]. Bagdasaryan et al. [8] and Pujol et al. [145] study disparity in accuracy under differential privacy (DP), and show that training with DP can increase disparate impact. In this work, we develop a theory that supports the empirical evidence that disparate impact would also cause disparity in vulnerability to MIAs [31, 118, 161].

4.8 Concluding Remarks

We have provided the first formal analysis of the disparate vulnerability of population subgroups to membership inference attacks. Our analysis provides new insights into why and when vulnerability to MIAs arises and why and when these attacks have disparate impact.

Key takeaways. The first key learning of our study is that fully preventing MIAs, and thus preventing disparate vulnerability can only be done in two ways. Either by significantly increasing the complexity of the learning problem to ensure distributional generalization; or using a differentially-private training algorithm with the associated side effects.

The second learning surfaces a more general problem: the consequences of the unreliability of privacy estimation for demographic groups with a minority representation in the data. We show that for small subgroups it is easy to incorrectly estimate their protection indirectly via aggregate privacy measures, or directly when not considering biases adequately.

Why disparate vulnerability is important. Disparate vulnerability has crucial legal and policy significance. Companies moving data between organizations or across borders face frictions designed to protect fundamental rights established by the approximately 140 countries with largely conceptually and textually similar privacy regulation around the world [73]. For example, moving data from Europe into a country with significant state surveillance apparatus, such as the United States, is difficult after the European Court of Justice's judgement in *Schrems II*. Other countries, such as several in South Asia, have established specific personal data localization laws [14]. As a consequence, there is growing interest in attempting to replace a direct trade in personal data with various forms of trade in models trained on this data.

Yet vulnerability of models to MIAs or other attacks compromising confidentiality might in some situations qualify models themselves as personal data [172]. The accountability principle in European data protection law places the onus on data controllers to demonstrate that a model should not be classified this way, for example through privacy-estimation techniques. Our study indicates there is a real risk of "privacy-washing", laundering a model with aggregate statistics that mask vulnerabilities of subgroups. It is true that prior work has also indicated that aggregate analysis can hide MIA vulnerability to attacks focusing on structurally vulnerable records [119]. However, this appears easier to dismiss as an acceptable residual leakage risk compared to disparate risks concerning members of salient minority groups, as in a liberal democracy, a regulator is more accountable towards these than towards a socially arbitrary selection of persons.

Open challenges. Our results also uncover a new challenge. It is difficult for auditors or regulators to practically inspect disparate vulnerability, because they might lack a sufficient number of examples relating to a minority group. When the subgroup data is scarce, our methods could be underpowered to detect disparity; however, not using the statistical tests and unbiased estimation methods from Section 4.4 risks flagging disparity always whenever subgroup data differ, devaluing the meaning of the estimate.

This points to a need for theoretical results that can be used as foundation in practical regulatory contexts. Theoretical results may be able to help regulators better ascertain the limits of metrics presented to them, and the conditions under which a model is structurally likely to be vulnerable to different types of privacy attacks even without difficult-to-obtain empirical evidence. The initial results provided in this chapter can already significantly contribute to discussions around the classification of machine learning systems in relation to their risk of data leakage as business practices of using models to transport information continue to evolve.

Chapter 5

Realistic Adversarial Modeling for Tabular Data

This chapter is based on a peer-reviewed article entitled “Adversarial Robustness for Tabular Data through Cost and Utility Awareness” [97] by Klim Kireev, Bogdan Kulynych, and Carmela Troncoso, published in the Proceedings of 2023 Network and Distributed System Security (NDSS) Symposium.

5.1 Introduction

Adversarial examples are inputs deliberately crafted by an adversary to cause a classification mistake. They pose a threat in applications for which such mistakes can have a negative impact on deployed models (e.g., a financial loss [68] or a security breach [47, 75, 101]). Adversarial examples also have positive uses. For instance, they offer a means of redress in applications in which classification causes harm to its subjects (e.g., privacy-invasive applications [2, 91, 105]). We review a standard approach to formalizing adversarial examples in Section 1.4.

The literature on adversarial examples largely focuses on image [28, 71, 123, 128, 138, 168] and text domains [60, 111, 115, 179, 193]. Yet, many of the applications where adversarial examples are most damaging or helpful are not images or text. High-stakes fraud and abuse detection systems [29], risk-scoring systems [68], operate on *tabular data*: A cocktail of categorical, ordinal, and numeric features. As opposed to images, each of these features has its own different semantics. For example, in a typical representation of an image, all dimensions of an input vector are similar in their semantics: they represent a color of a pixel. In tabular data, one dimension could correspond to a numeric value of a person's salary, another to their age, and another to a categorical value representing their marital status. The properties of the image domain have shaped the way adversarial examples and adversarial robustness are approached in the literature [128] and have greatly influenced adversarial robustness research in the text domain. In this chapter, we argue that adversarial examples in tabular domains are of a different nature, and adversarial robustness has a different meaning. Thus, the definitions and techniques used to study these phenomena need to be revisited to reflect the tabular context.

We argue that two high-level differences need to be addressed. First, imperceptibility, which is the main requirement considered for image and text adversarial examples, is ill-defined and can be irrelevant for tabular data. Second, existing methods assume that all adversarial inputs have the same value for the adversary, whereas in tabular domains different examples can bring drastically different gains.

Imperceptibility and semantic similarity are not necessarily the primary constraints in tabular domains. The existing literature commonly formalizes the concept of "an example deliberately crafted to cause a misclassification" as a *natural example*, i.e., an example coming from the data distribution, that is *imperceptibly* modified by an adversary in a way that the classifier's decision changes. Typically, imperceptibility is formalized as closeness according to a mathematical distance such as L_p [160, 200].

In tabular data, however, imperceptibility is not necessarily relevant. Let us consider the following toy example of financial-fraud detection: Assume a fraud detector takes as input two features: (1) transaction amount, and (2) device from which the transaction was sent. The adversary aims to create a fraudulent financial transaction. The adversary starts with a natural example (amount=$200, device=‘Android phone’) and changes the feature values until the detector no longer classifies the example as fraud. In this example, *imperceptibility is not well-defined.* Is a modification to the amount feature from $200 to $201 imperceptible? What increase or decrease would we consider perceptible? The issue is even more apparent with categorical data, for which standard distances such as L_2, L_∞ cannot even capture imperceptibility: Is a change of the device feature from Android to an iPhone imperceptible? Even if imperceptibility was well-defined, *imperceptibility might not be relevant.* Should we only be concerned about adversaries making “imperceptible” changes, e.g., modifying amount from $200 to $201? What about attack vectors in which the adversary evades detection while changing the transaction by a “perceptible” amount: from $200 to $2,000?

Formalizing adversarial examples as imperceptible modifications narrows the mathematical tools that can be used to study adversarial examples in their broad sense. In the case of tabular data, this prevents the study of techniques that adversaries could employ in “perceptible”, yet effective ways.

We argue that in tabular data the primary constraint should be *adversarial cost*, rather than any notion of similarity. Instead of looking at how visually or semantically similar are the feature vectors, the focus should be on *how costly it is for an adversary to enact a modification.* Costs capture the effort of the adversary, e.g., financial or computational. “How much money does the adversary have to spend to evade the detector?” better captures the possibility that an adversary deploys an attack than establishing a threshold on the L_p distance the adversary could tolerate. In the fraud-detection example, regardless of whether a change from Android to iPhone is imperceptible and semantically similar or not, it is certain that the change costs the adversary a certain amount of resources. How significant are these costs determines the likelihood of the adversary deploying such an attack.

Different tabular adversarial examples are of different value to the adversary. In the literature, with a notable exception of Zhang and Evans [201], defenses against adversarial examples implicitly assume that all adversarial examples are equal in their importance [71, 158, 187, 198]. In tabular data domains, however, different adversarial examples can bring very different *gains* to the adversary. In the fraud-detection example, if a fraudulent transaction with transaction amount of $2,000 successfully evades the detector, it could be significantly more profitable to the adversary than a transaction with amount of $200.

Using the adversarial cost as the primary constraint for adversarial examples provides a natural way to incorporate the variability in adversarial gain. The adversary is expected to care about the profit obtained from the attack, i.e., the difference between the cost associated with crafting an adversarial example, and the gain from its successful deployment. We call this difference the *utility* of the attack. We show how utility can be incorporated into the design of attacks to ensure their economic profitability, and into the design of defenses to ensure protection against adversaries that focus on profit.

In this chapter, we introduce a framework to study adversarial examples tailored to tabular data. Our contributions:

- We propose two *adversarial objectives* for tabular data that address the limitations of the standard approaches: a *cost-bounded* objective that substitutes standard imperceptibility constraints with adversarial costs; and a novel *utility-bounded* objective in which the adversary adjusts their expenditure on different adversarial examples proportionally to the potential gains from deploying them.
- We propose a practical attack algorithm based on greedy best-first graph search for crafting adversarial examples that achieve the objectives above.
- We empirically evaluate our attacks in realistic conditions demonstrating their applicability to real-world security scenarios, showing that these attacks can bring about utility to the adversary.

In summary, this chapter presents a framework for studying adversarial robustness that is specifically designed for tabular data from ground-up.

5.2 Evasion Attacks

This section introduces the notation and the formal setup of *evasion attacks* [see, e.g., 15, 137] in tabular domains, which is closely related but differs from the standard setting of adversarial robustness introduced in Section 1.4.

Feature space in tabular domains. The input domain's *feature space* $\mathbb{X}$ is composed of m features: $\mathbb{X} \subseteq \mathbb{X}_1 \times \mathbb{X}_2 \times \cdots \times \mathbb{X}_m$. For example $x \in \mathbb{X}$, we denote the value of its i-th feature as x_i. Features x_i can be categorical, ordinal, or numeric. Each example is associated with a binary class label $y \in \{0, 1\}$.

Target classifier. We assume the adversary's *target* to be a binary classifier $f_\theta(x) \in \{0, 1\}$, with a confidence score function $h_\theta(x) \in [0, 1]$. We omit the θ subscript in this chapter for conciseness. We focus on binary classification as it is the task in which adversarial dynamics typically arise in tabular domains (e.g., fraud detection [29] or risk-scoring systems [68]).

Adversarial examples. An evasion attack proceeds as follows: The adversary starts with an initial example $x \in \mathbb{X}$ with a label $y = y_s$. We call this class the *adversary's* *source class.* The adversary's goal is to modify x to produce an *adversarial example* x^* that is classified as $f(x^*) = y_t$, $y_s \neq y_t$. We call this the *adversary's target class.* The attack is *successful* if the adversary can produce such an adversarial example. Depending on the adversarial objective, the adversarial example might also need to satisfy additional constraints, as detailed in Section 5.3.

Because an attack is performed using an adversarial example, as in the literature, we use the terms *adversarial example* and *attack* interchangeably.

Our methods can be used in a multi-class setting as they are agnostic to which class is the target one. Our notation, however, is specific to the binary setting for clarity.

Adversarial model. In terms of capabilities, we assume the adversary can only perform modifications that are within the domain constraints. In the fraud-detection example, the adversary can change the transaction amount, but the value must be positive. For a given initial labeled example (x, y), we denote the set of feasible adversarial examples that can be reached within the capabilities of the adversary as $\mathcal{F}(x, y) \subseteq \mathbb{X}$.

In terms of knowledge, we assume that the adversary has black-box access to the target classifier: The adversary can issue queries using arbitrary examples and obtain $h(x)$. In our evaluation (Section 5.5) we compare this adversary against existing attacks with white-box access to the gradients.

Preservation of semantics. It is common to require that an adversarial example is *semantics-preserving* [142, 160]: the adversarial example retains the same true class as the original example. We do not impose such a requirement. The only constraint we impose is that the modifications leading to an adversarial example *are feasible within the domain constraints*, i.e., that the adversarial example belongs to the set $\mathcal{F}(x, y)$. This is because in tabular domains limiting $\mathcal{F}(x, y)$ to those adversarial examples that also preserve semantics is counterproductive: As long as the adversary successfully achieves their goal with an adversarial example that is feasible and is within their budget (see Section 5.3), the attack presents a valid threat.

5.3 Adversarial Objectives in Tabular Data

As we detail in Section 5.1, the approaches to adversarial modeling tailored to image or text data have two critical limitations when applied to tabular domains:

1. *Focus on imperceptibility and semantic similarity.* Neither closeness to natural examples in L_p distance, nor closeness in terms of semantic similarity, is a

well-applicable definition of adversarial examples in tabular domains. This is because such similarities are either ill-defined for mixed-type features (e.g., as is the case with L_p distance), or potentially irrelevant to the quantification of adversarial constraints (both L_p and semantic similarity).

2. *Assuming all adversarial examples are equally useful.* Most existing defenses against adversarial examples do not distinguish different attacks in terms of their value for the adversary. In tabular domains, due to the inherent heterogeneity of the data, some attacks could bring significantly more gain to the adversary.

Next, we propose adversarial objectives which aim to address these limitations.

5.3.1 Cost-Bounded Objective

Evasion attacks which use adversarial costs were first formalized in early works on adversarial machine learning [12, 121]. In these works, the adversary aims to find evading examples with minimal cost. Since the discovery of adversarial examples in computer vision models [168], this formalization was largely abandoned in favor of constraints based on L_p and other mathematical distances (e.g., Wasserstein distance [188] or LPIPS [96, 110]). In this work, we revisit the cost-oriented approach, which better reflects the adversary's capabilities in tabular domains.

In a standard way to obtain an adversarial example (see Section 1.4), the adversary aims to construct an example that maximizes the classification loss incurred by the target classifier $f_\theta(x)$, while keeping the L_p-distance from the initial example bounded:

$$\max_{x' \in \mathcal{F}(x,y)} \ell((x', y); \theta) \quad \text{s.t. } \|x' - x\|_p \leq \varepsilon \tag{5.1}$$

This objective implicitly assumes that the adversary wants to keep the adversarial example as similar to the initial example as possible in terms of the examples' feature values. The closeness in terms of L_p distance aims to capture imperceptibility and to preserve the original example's semantics [160].

From Distances to Costs. To address the fact that imperceptibility or semantic similarity is not necessarily relevant for adversarial settings in tabular domains, we adapt the definition in Eq. (5.1) to the tabular setting by introducing a cost constraint.

This constraint represents the limited amount of resources available to the adversary to evade the target classifier. If the adversary can find an adversarial example that achieves this goal within the cost budget, the adversary proceeds with the attack. Formally, we associate a cost to the modifications needed to generate any adversarial example $x' \in \mathcal{F}(x, y)$ from the original example (x, y). We encode this cost as a function $c : \mathbb{X} \times \mathbb{X} \to \mathbb{R}^+$. We assume the generation cost is zero if and only if no change is enacted: $c(x, x') = 0 \iff x = x'$.

This formulation is generic: it can encompass geometric and semantic distances, but it goes beyond that. It exhibits the following desirable properties:

a. *Support for arbitrary feature types and rich semantics.* Whereas L_p distances only support numeric features, our generic cost model can support any feature type. This is because it does not enforce any structural constraints on the exact form of cost of changing a feature value x_i into x'_i. For example, the cost does not need to obey $|x_i - x'_i|$ as would be the case with L_1 distance. Moreover, unlike mathematical distances, our model does not require the costs to be symmetric. For instance, an increase in a feature value could have a different cost than a decrease.

b. *Enables more generic quantification of adversarial effort.* Our cost model imposes neither a geometric structure such as is the case with L_p distances, nor any ties to semantic similarity. Thus, the costs can be quantified in those units that are directly relevant to adversarial constraints. An important use case is that our model supports defining costs in the financial sense, i.e., assigning a dollar cost to mounting an attack with a given adversarial example as opposed to semantic closeness or closeness in feature space.

c. *Support for feature-level accumulation.* Related literature on attacks in tabular data often formalizes costs using indepenent per-feature constraints (see Section 5.6). Although our generic cost model supports such a special case, it also enables accumulation of per-feature costs. Therefore, it can encode a realistic assumption that changing more features increases adversary's expenditure.

The optimization problem. We assume that the cost-bounded adversary has a budget ε. The adversary aims to find any example that flips the classifier's decision *and* that is within the cost budget:

$$\max_{x' \in \mathcal{F}(x)} \mathbb{1}[f_\theta(x') \neq y] \quad \text{s.t. } c(x, x') \leq \varepsilon \tag{5.2}$$

Alternatively, the adversary can optimize a standard surrogate objective which ensures that the optimization problem can be solved in practice:

$$\max_{x \in \mathcal{F}(x,y)} \ell((x, y); \theta) \quad \text{s.t. } c(x, x') \leq \varepsilon, \tag{5.3}$$

In the surrogate form, the optimization problem of the cost-bounded adversary is an adaptation of Eq. (5.1) with the norm constraint substituted by the adversarial-cost constraint. This formalization is in line with recent formalizations of adversarial examples [123], as opposed to early approaches which aim to find minimal-cost attacks [121].

5.3.2 Utility-Bounded Objective

The cost-bounded adversarial objective solves the issue of imperceptibility and semantic similarity not being suitable constraints for tabular data. It does not, however, tackle the problem of heterogeneity of examples: the adversary cannot assign different importance to different adversarial examples. In a realistic environment, it can be a serious drawback. For instance, an adversary might spend more resources than they gain from a successful attack. Another instance is the defender hypothetically suffering serious losses due to high-impact adversarial examples, even if for the majority of examples the defense is appropriate.

We propose to capture this heterogeneity by introducing the *gain* of an attack. The gain, $r : \mathbb{X} \to \mathbb{R}^+$, represents the reward (e.g., the revenue) that the adversary receives if their attack using a given adversarial example is successful.

We also introduce the concept of *utility*: the net benefit of deploying a successful attack. We define the utility $u_{x,y}(x^*)$ of an attack mounted with adversarial example x^* as simply the gain minus the costs:

$$u_{x,y}(x^*) \triangleq r(x^*) - c(x, x^*), \tag{5.4}$$

where (x, y) is the initial example.

Recall that the adversary has black-box access to the target classifier. Thus, they can learn whether an example x^* evades the classifier or not (i.e., whether $f(x^*) \neq y$). Then, they can decide to deploy an attack with an adversarial example x^* only if the utility of the attack exceeds a given *margin* $\tau \geq 0$. Otherwise, the adversary discards this adversarial example. Formally, we can model this process by using a *utility constraint* instead of a cost constraint:

$$\max_{x \in \mathcal{F}(x,y)} \mathbb{1}[f(x) \neq y] \quad \text{s.t. } u_{x,y}(x) \geq \tau \tag{5.5}$$

If we assume that the gain is constant for any adversarial example $x' \in \mathcal{F}(x, y)$ that is a modification of an initial example (x, y), that is, $r(x) = r(x')$, this problem can also be seen as a variant of the cost-bounded formulation in Eq. (5.2), where ε varies for different initial examples:

$$\begin{aligned}
&\max_{x' \in \mathcal{F}(x,y)} \mathbb{1}[f(x') \neq y] \\
&\quad \text{s.t. } u_{x,y}(x') \geq \tau \\
&\iff r(x) - c(x, x') \geq \tau \\
&\iff c(x, x') \leq \varepsilon(x) \triangleq r(x) - \tau
\end{aligned} \tag{5.6}$$

In Appendix E.1, we discuss the formalization of a utility-maximization objective which models an adversary which wants to maximize their profit subject to budget constraints.

5.3.3 Quantifying Cost and Utility

A natural question in our setup is how to define the adversary's costs and gains. This question is relevant to all related prior work on adversarial robustness in tabular data (see Section 5.6). For example, if adversarial robustness is defined in terms of an L_p distance, both the attacker and defender need to determine an acceptable perturbation magnitude, which inherently comes from domain knowledge.

In our applications (see Section 5.5), we focus on the settings in which adversarial capabilities are constrained in terms of financial costs. In such settings, we expect that the adversary is able to quantify the financial costs $c(x, x')$ and gains $r(x)$ by practical necessity. On the defender's side, estimating these values is trickier, as the defender might be unaware of the exact capabilities of the adversary. The defender thus needs to employ standard threat modeling techniques and domain knowledge. It is worth mentioning that the defender is not required to estimate the capabilities perfectly. Rather, they need to obtain the lower bound on the adversary's costs. After that, if the defended system is robust, it is robust against the adversary whose costs are at least as high as estimated.

In our utilitarian approach, it is possible to include other concerns and constraints of the adversary as part of the utility definition by measuring them in the same units as the utility (e.g., financial costs). For instance, as the driving concern behind imperceptibility-based approaches to adversarial robustness is the detection of an attack, the gain could be adjusted for a potential risk of being detected. The adversary could estimate the probability of being detected (e.g. using public statistics), and incorporate it into the gain by subtracting an expected value of the attack failure due to detection.

5.4 Finding Adversarial Examples in Tabular Domains

In this section, we propose practical algorithms for finding adversarial examples suitable to achieve the adversarial objectives we introduce in Section 5.3.

5.4.1 Graphical Framework

The optimization problems in Section 5.3 can seem daunting due to the large cardinality of $\mathcal{F}(x, y)$ when the feature space is large. To make the problems tractable, we transform them into graph-search problems, following the approach by Kulynych et al. [104]. Consider an example-specific *state-space graph* parameterized by (V, E). Each node corresponds to a feasible example in the feature space, $V = \mathcal{F}(x, y) \cup \{x\}$. Edges between two nodes x and x' exist if and only if they differ in value of one feature: there exists $i = 1, \ldots, n$ such that $x_i \neq x'_i$, and $x_j = x'_j$ for all $j \neq i$. In other words, the immediate descendants of a node in the graph consist of all feasible feature vectors that differ from the parent in exactly one feature value.

Using this state-space graph abstraction, the objectives in Section 5.3 can be modeled as graph-search problems. Even though the graph size is exponential in the number of feature values, the search can be efficient. This is because it can construct the relevant parts of the graph on the fly as opposed to constructing the full graph in advance.

Building the state-space graph is straightforward when features take discrete values. To encode continuous features in the graph we discretize them by only considering changes to a continuous feature i that lie within a finite subset of its domain $\mathbb{X}_i$, in particular, on a discrete grid. The search efficiency depends on the size of the grid. As the grid gets coarser, finding adversarial examples becomes easier. This efficiency comes at the cost of potentially missing adversarial examples that are not represented on the grid but could fulfil the adversarial constraints with less cost or higher utility.

5.4.2 Attacks as Graph Search

In the remainder of the chapter, we make the following assumptions about the adversarial model:

Assumption 5.1 (Modular costs)**.** The adversary's costs are *modular*: they decompose by features. Formally, changing the value of each feature i from x_i to x'_i has the associated cost $c_i(x_i, x'_i) > 0$, and the total cost of modifying x into x' is a sum of

Algorithm 6 Best-First Search (BFS)

```
function BFS$_{B,s,\varepsilon}(x)$
    open ← MinPriorityQueue$_B(x, 0)$
    closed ← $\{\}$
    while open is not empty do
        $v$ ← open.pop()
        if $v \notin$ closed then
            closed ← closed $\cup \{v\}$
        if $\eta(v) \geq \delta$ then return $v$
        $S$ ← expand($v$)
        for $t \in S$ do
            if $t \notin$ closed and $c(x, t) \leq \varepsilon$ then
                open.add($t, s(v, t)$)
```

individual feature-modification costs:

$$c(x, x') = \sum_{i}^{n} c_i(x_i, x'_i) \tag{5.7}$$

The state-space graph can encode modular costs by assigning weights to the graph edges. An edge between x and x' has an associated weight of $c_i(x_i, x'_i)$, where i is the index of the feature that differs between x and x'. For pairs of examples $x^{(0)}$ and $x^{(t)}$ that differ in more than one feature, the cost $c(x^{(0)}, x^{(t)})$ is the sum of the edge costs along the shortest path from $x^{(0)}$ to $x^{(t)}$.

Assumption 5.2 (Constant gain)**.** For any initial example (x, y), the adversary cannot change the gain:

$$\forall x' \in \mathcal{F}(x, y): \quad r(x) = r(x') \tag{5.8}$$

This follows the approach in utility-oriented strategic classification (as detailed in Section 5.3.2). This assumption is not formally required for our attack algorithms (described next in this section), but we focus on this setting in our empirical evaluations.

Strategies to find adversarial examples. Under our two assumptions, the cost-bounded objective in Eq. (5.2) and the utility-bounded objective in Eq. (5.6) can be achieved by finding any adversarial example that is classified as target class and is within a given cost bound. Thus, these adversarial goals can be achieved using *bounded-cost search* [165].

We start with the *best-first search* (BFS) [81, 104], a flexible meta-algorithm that generalizes many common graph search algorithms. In its generic version (Algorithm 6) BFS keeps a bounded priority queue of *open nodes*. It iteratively pops the node v with the highest score value from the queue (best first), and adds its immediate descendants to the queue. This is repeated until the queue is empty. The algorithm returns the node with the highest score out of all popped nodes.

The BFS algorithm is parameterized by the *scoring function* $s : V \times V = \mathbb{X} \times \mathbb{X}$ and the size of the priority queue B. Different choices of the scoring function yield search algorithms suited for solving different graph-search problems, such as Potential Search for bounded-cost search [165, 166], and A* [46, 102] for finding the minimal-cost paths. When $B = \infty$, the algorithm might traverse the full graph and is capable of returning the optimal solution. As the size of B decreases, the optimality guarantees are lost. When $B = 1$ BFS becomes a *greedy* algorithm that myopically optimizes the scoring function. When $1 < B < \infty$ we get a *beam search* algorithm that keeps B best candidates at each iteration.

To achieve the adversarial objectives in Section 5.3, we propose to use a concrete instantiation of BFS, what we call the *Universal Greedy (UG)* algorithm. Inspired by heuristics for cost-bounded optimization of submodular functions [95, 186], we set the scoring function to balance the increase in the classifier's score and the cost of change:

$$s(v, t) = -\frac{h(t) - h(v)}{c(v, t)} \tag{5.9}$$

The minus sign appears because BFS expands the lowest scores first, and we need to maximize the score. We set the beam size to $B = 1$ (greedy), which enables us to find high-quality solutions to *both* cost-bounded and utility-bounded problems at reasonable computational costs (see Section 5.5).

5.5 Experimental Evaluation

In this section, we show that our graph-based attacks can be used by adversaries to obtain profit, and that our proposed defenses are effective at mitigating damage from these attacks.

5.5.1 Setup

Datasets. We perform our experiments on three tabular datasets which represent real-world applications for which adversarial examples can have social or economic implications:

- **TwitterBot** [69]. The dataset contains information about more than 3,400 Twitter accounts either belonging to humans or bots. The task is to detect bot accounts. We assume that the adversary is able to purchase bot accounts and interactions through darknet markets, thus modifying the features that correspond to the account age, number of likes, and retweets.

- **IEEE-CIS**[1]. The dataset contains information about around 600K financial transactions. The task is to predict whether a transaction is fraudulent or benign. We model an adversary that can modify three features for which we can outline the hypothetical method of possible modification, and estimate its cost: payment-card type, email domain, and payment-device type.

- **HomeCredit**[2]. The dataset contains financial information about 300K home-loan applicants. The main task is to predict whether an applicant will repay the loan or default. We use 33 features, selected based on the best solutions to the original Kaggle competition. Of these, we assume that 28 can be modified by the adversary, e.g., the loan appointment time.

Models. We evaluate our attacks against three types of ML models commonly applied to tabular data. First, an L_2-regularized *logistic regression (LR)* with a regularization parameter chosen using 5-fold cross-validation. Second, *XGBoost gradient-boosted decision trees (XGBT).* Third, *TabNet* [6], an attentive transformer neural network specifically designed for tabular data. We optimize the number of steps as well as the capacity of TabNet's fully connected layers using grid search.

Adversarially modifiable features. We assume that the feasible set consists of all positive values of numerical features and all possible values of categorical features. For simplicity, we avoid features with mutual dependencies and treat the adversarially modifiable features as independent. We detail the choice of the modifiable features and their costs in Appendix E.2.3.

Metrics. To evaluate the effectiveness of the attacks and defenses, we use three main metrics:

- *Adversary's success rate:* The proportion of correctly classified examples from a test dataset $\bar{S} \in \mathbb{D}^n$ for which adversarial examples successfully generated using the attack algorithm $\mathcal{A}(x, y)$ evade the classifier:

$$\Pr_{(x,y)\sim\bar{S}}[f(\mathcal{A}(x,y)) \neq y \wedge f(x) = y] \,.$$

[1] https://www.kaggle.com/c/ieee-fraud-detection

[2] https://www.kaggle.com/c/home-credit-default-risk

- *Adversarial cost:* Average cost of successful adversarial examples:

$$\underset{(x,y)\sim\bar{S}}{\mathbb{E}}\left[c(x,\mathcal{A}(x,y)) \mid f(\mathcal{A}(x,y)) \neq y \wedge f(x) = y\right].$$

- *Adversarial utility:* Average utility (see Eq. (5.4)) of successful adversarial examples:

$$\underset{(x,y)\sim\bar{S}}{\mathbb{E}}[u_{x,y}(\mathcal{A}(x,y)) \mid f(\mathcal{A}(x,y)) \neq y \wedge f(x) = y].$$

In all cases, we only consider correctly classified initial examples which enables us to distinguish these security metrics from the target model's accuracy. We introduce additional metrics in the experiments when needed.

5.5.2 Design Choices of the Universal Greedy Algorithm

When designing attack algorithms in the BFS framework (see Algorithm 6), there are two main design choices: the scoring function, and the beam size. We explore different configurations and show that our parameter choices for the Universal Greedy attack produce high-quality adversarial examples.

Beam size. We define the beam size of the Universal Greedy attack to be one. The other options that we evaluate are 10 and 100. We evaluate them by running three types of attacks: cost-bounded for three cost bounds ε, and utility-bounded at the breakeven margin $\tau = 0$. We compute two metrics: attack success, and the success-to-runtime ratio. This ratio represents how much time is needed to achieve the same level of success rate using each choice of the beam size. This metric is more informative for our evaluation than runtime, as the runtime is simply proportional to the beam size.

For feasibility reasons, we use two datasets: TwitterBot and IEEE-CIS. We aggregate the metrics across the three models (LR, XGBT, TabNet), and report the average. The results on TwitterBot are equivalent to the results on IEEE-CIS, thus for conciseness we only report IEEE-CIS results.

We find that the success rates are equal up to the percentage point for all choices of the beam size. We show the detailed numeric results in Table E.5 in the Appendix. As the smallest beam size of one is the fastest to run, it demonstrates the best success/time ratio, therefore, is the best choice.

Scoring function. Recall from Eq. (5.9) that the scoring function is the cost-weighted increase in the target classifier's confidence, which aims to maximize the increase in classifier confidence at the lowest cost.

Other choices for the scoring function $s(v, t)$ could be:

- A^* *algorithm* [46, 102, 104]: $s(v, t) = c(v, t) + \lambda \cdot \chi(t)$, where $\chi(t)$ is a heuristic function, which estimates the remaining cost to a solution and $\lambda > 0$ is a greediness parameter [143]. This scoring function balances the current known cost of a candidate and the estimated remaining cost. We choose the model's confidence for the positive class, $\chi(x) = h(x)$, as a heuristic function. Intuitively, this works as a heuristic, because the lower the confidence for the positive class, the more likely we are close to a solution: an example classified as the target class.
- *Potential Search* (PS) [165, 166]: $s(v, t) = {}^{\chi(t)}\!/\!_{(\varepsilon - c(v,t))}$, which additionally takes into account the cost bound ε, thus becoming more greedy (i.e., optimizing $s(v, t) = \lambda \cdot \chi(t)$ with $\lambda \approx {}^{1}\!/\!_{\varepsilon}$) when the cost of the current candidate leaves a lot of room within the ε budget. We also choose $\chi(x) = h(x)$ as a heuristic function.
- *Basic Greedy*: $s(v, t) = {}^{-h(t)}\!/\!_{c(s,t)}$, which aims to maximize the classifier's confidence, yet balance it with the incurred cost. Unlike Eq. (5.9), this scoring function does not take into account the relative increase of the confidence, only its absolute value.

We evaluate the choice of the scoring function on the TwitterBot and IEEE-CIS datasets, with the beam size fixed to one. We run the cost-bounded and utility-bounded attacks in the same configuration as before, and measure two metrics averaged over the models: Attack success, and attack success/time ratio.

Table 5.1 shows the results. On IEEE-CIS, the Universal Greedy outperforms the other choices in terms of success rate and the success/time ratio. On the TwitterBot dataset, it outperforms the other choices in the utility-bounded and unbounded attacks. For cost-bounded attacks, the Universal Greedy offers very close performance to the best option, the Basic Greedy.

5.5.3 Graph-Based Attacks vs. Baselines

We compare the Universal Greedy (UG) algorithm against two baselines: previous work, and the minimal-cost adversarial examples.

Previous Work: PGD. As our cost model differs from the existing approaches to attacks on tabular data, we fundamentally cannot perform a fully apples-to-apples comparison against existing attacks (see Section 5.4). To compare against the high-level ideas from prior work, we follow the spirit of the attack by Ballet et al. [9], which modifies the optimization problem from Eq. (5.1) to use correlation-based weights.

Table 5.1: *Effect of the scoring-function choice* for graph-based attacks. In the majority of settings, our Universal Greedy scoring function offers the best success rate and performance.

(a) IEEE-CIS

	Adv. success, %			
Cost bound →	10	30	Gain	∞
Scoring func. ↓				
UG	**45.32**	**56.57**	**56.22**	**68.20**
A*	42.37	55.62	55.34	53.47
PS	**45.32**	55.14	56.18	N/A
Basic Greedy	42.37	55.46	55.38	53.82

	Success/time ratio			
Cost bound →	10	30	Gain	∞
Scoring func. ↓				
UG	**3.78**	**4.80**	**2.53**	**2.06**
A*	3.29	3.83	1.89	1.15
PS	**3.78**	4.01	2.26	N/A
Basic Greedy	3.21	3.86	2.01	1.16

(b) TwitterBot

	Adv. success, %			
Cost bound →	1,000	10,000	Gain	∞
Scoring func. ↓				
UG	80.24	**85.35**	**21.63**	**87.00**
A*	77.56	84.45	20.29	86.25
PS	79.95	85.19	21.48	N/A
Basic Greedy	**80.40**	85.04	**21.63**	86.85

	Success/time ratio			
Cost bound →	1,000	10,000	Gain	∞
Scoring func. ↓				
UG	208.95	205.76	**64.99**	**205.31**
A*	206.33	201.93	62.25	201.31
PS	205.85	203.18	63.76	N/A
Basic Greedy	**210.20**	**206.20**	64.32	204.96

Thus, we adapt the standard PGD attack (see Section 1.4) to (1) support categorical features through discretization, and (2) use weighted L_1 norm to support per-feature costs. We provide a detailed description of this adaptation in Algorithm 7. In this adaptation, we use projection onto a w-weighted L_1 ball $\mathsf{Proj}_{x,p,w,\varepsilon}(x')$, defined analogously to the standard projection (see Section 1.4):

$$\mathsf{Proj}_{x,p,w,\varepsilon}(x') \triangleq \min_{\bar{x}\in\mathbb{R}^d} \|\bar{x} - x'\|_2 \quad \text{s.t.} \quad \|\bar{x} - x\|_{p,w} \leq \varepsilon. \tag{5.10}$$

We use an algorithm for efficient projection onto a weighted L_1 ball by Perez et al. [141]. We use the vector of weights to encode per-feature costs. This encoding, however, fundamentally cannot represent all possible cost functions we support (see Section 5.4.2). We nevertheless use it as we aim to provide a best-effort comparison to prior work.

We run attacks using PGD with 100 and 1,000 steps, and compare it to UG (Section 5.4) on the TwitterBot and IEEE-CIS datasets. As PGD can only operate on differentiable models, in this comparison we only evaluate the performance of the attacks against TabNet.

We run the cost-bounded attacks using two values of the ε bound, specific to each dataset (see Appendix E.2 for the exact attack parameters). As before, we also run a utility-bounded attack at the breakeven margin $\tau = 0$. We measure the success rates of the attacks, as well as the average cost of the obtained adversarial examples. For

Algorithm 7 PGD-Based Attack

```
Input: Initial example $(x, y)$, weight vector $w \in \mathbb{R}^m$, cost bound $\varepsilon$, number of iterations $t_{\max}$
Output: Adversarial example $x_{t_{\max}}$
$\alpha \leftarrow 2\frac{\varepsilon}{t_{\max}}$
for $t$ in $1, \ldots, t_{\max}$ do
    $\mathsf{grad}_t \leftarrow \nabla_\delta \ell((x + \delta, y); \theta)$
    $x_t = \mathsf{Proj}_{x,p,w,\varepsilon}(x_{t-1} + \alpha \frac{\mathsf{grad}_t}{\|\mathsf{grad}_t\|_1})$
```

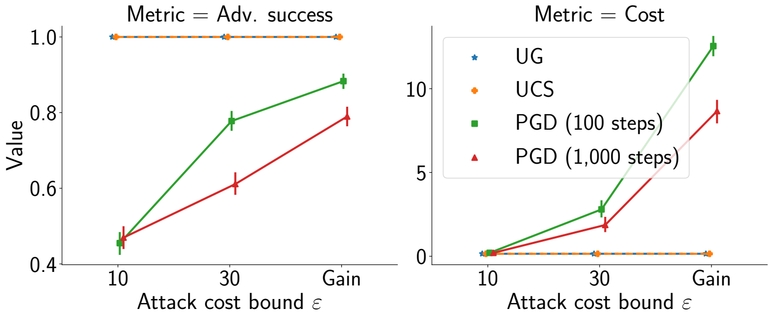


Figure 5.1: *Universal Greedy attack vs Baselines.* Left: Attack success rate (higher is better for the adversary). Right: Attack cost (lower is better for the adversary). For all cost bounds, our graph-based attack outperforms standard PGD and returns close to optimal-cost adversarial examples (obtained with Uniform-Cost Search, UCS).

conciseness, we do not report the results on TwitterBot, as they find they are equivalent to those on IEEE-CIS.

Fig. 5.1 shows that the UG attack consistently outperforms the PGD-based baseline both in terms of the success rate and the costs. Our attacks are superior even when the PGD-based baseline produces feasible adversarial examples.

Minimal-Cost Adversarial Examples. As UG is a greedy algorithm, we additionally evaluate how far are the obtained adversarial examples from the optimal ones in terms of cost. For this, we compare the results from UG to a standard Uniform-Cost Search (UCS) [104]. UCS is an instantiation of the BFS framework (see Section 5.4) with unbounded beam size, and the scoring function equal to the cost: $s(v, t) = c(v, t)$. In our setting, UCS is guaranteed to return optimal solutions to the following optimization problem:

$$\min_{x' \in \mathcal{F}(x,y)} c(x, x') \qquad \text{s.t. } f(x') \neq y \tag{5.11}$$

Fig. 5.1 shows that UG has almost no overhead over the minimal-cost adversarial examples on TabNet ($1.03\times$ overhead on average). In fact, the average and median cost overhead is $1.80\times$ and $1\times$ over all models, respectively. There exist some outlier examples, however, with over $100\times$ cost overhead. We provide more information on the distribution of cost overhead in Appendix E.2.

5.5.4 Attack Performance

Having shown that the attacks outperform the baseline, and the design choices are sound, we demonstrate that the attacks bring some *utility* to the adversary. In this section, we evaluate the attacks in a non-strategic setting: the models are not deliberately defended against the attacks. For conciseness, we only evaluate cost-bounded attacks, as the next section provides an extensive demonstration of utility-bounded attacks.

In *all* evaluated settings, the attacks have non-zero success rates and achieve non-zero adversarial utility. Fig. 5.2 show the results of cost-bounded attacks for IEEE-CIS and HomeCredit datasets. We omit the results for LR on HomeCredit as this model does not perform better than the random baseline. An average adversarial example obtained using the cost-bounded objective brings a profit of $125 to the adversary when attacking the IEEE-CIS TabNet model, and close to 100% of examples in the test data can be turned into successful adversarial examples.

Although for all models we see non-zero success and utility, some models are less vulnerable than others, even without any protection. For example, the success rate of the adversary against LR on IEEE-CIS is much lower than against TabNet (at least 50 p.p. lower). This model, however, is also comparatively inaccurate, with only 62% classification accuracy.

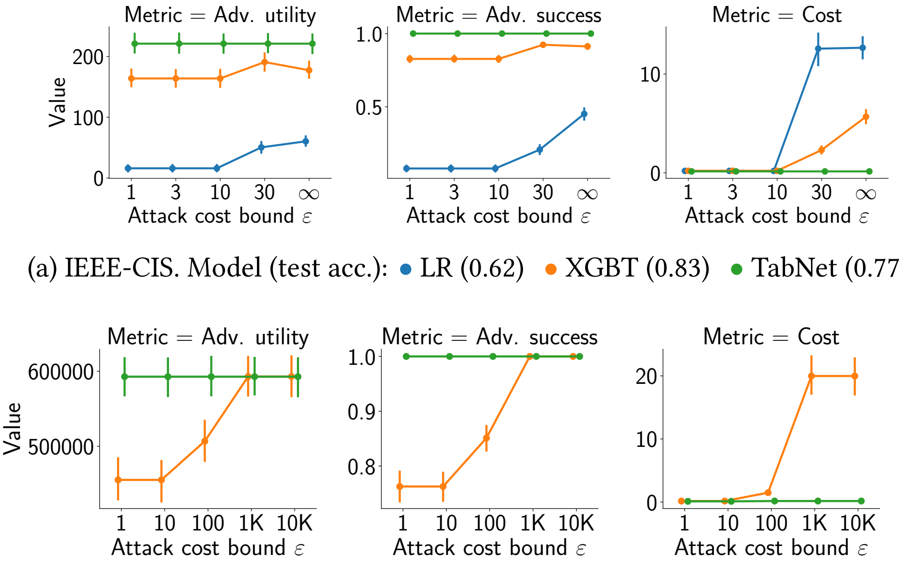
5.6 Related Work

Our conceptual contributions span three aspects of adversarial robustness in tabular domains: new formulations of *adversarial objectives* and *attack strategies* within these objectives. We review the related work in each of the aspects next. We also provide a concise summary in Table 5.2.

5.6.1 Adversarial Objectives

In this part, we review the related adversarial objectives as well as some approaches which are similar in spirit to our adversarial objectives.

Cost-based objectives. Our generic cost-bounded objective is not the only possible



(a) IEEE-CIS. Model (test acc.): ● LR (0.62) ● XGBT (0.83) ● TabNet (0.77)

(b) HomeCredit. Model (test acc.): ● XGBT (0.65) ● TabNet (0.68)

Figure 5.2: Results of cost-bounded graph-based attacks against three types of models. Left pane: Adversarial utility (higher is better for the adversary). Middle and right panes: See Fig. 5.1. On IEEE-CIS, the attack can achieve utility from approximately $10 to $125 per attack depending on the target model. On HomeCredit, the average utility ranges between $400,000 and $600,000.

Table 5.2: Summary of related work in terms of three aspects: adversarial models, attack strategies, and defense strategies. Adversarial models: *Adv. cost* — description of (an equivalent of) an adversarial cost model. *Adv. utility* — whether adversarial gain is incorporated into the model. Attacks: *Targets* — which target models can be attacked. *Feasibility* — whether the attack is guaranteed to produce a feasible adversarial example. *Algorithm* — a short description of the algorithmic approach. Defenses: *Arch.* — which model architectures are supported.

	Adversarial models		Attack strategies		
	Adv. cost	Adv. utility	Targets	Feasibility	Algorithm
Ballet et al. [9]	Feature-importance based	—	Differentiable	✗	Gradient-based
Cartella et al. [30]	Feature-importance based	—	**Any**	✗	ZOO
Mathov et al. [125]	Distance based	—	**Any**	✗	Gradient-based
Kantchelian et al. [93]	L_p	—	Tree-based	✓	MILP
Andriushchenko and Hein [5]	L_∞	—	Tree-based	✓	Custom
Chen et al. [35]	Per-feature constraints	—	—	—	—
Calzavara et al. [26]	Per-feature constraints	—	Tree-based	✓	Exhaustive search
Vos and Verwer [174]	Per-feature constraints	—	—	—	—
Ours	**Generic** (Section 5.4.2)	✓	**Any**	✓	Graph search

approach to model attacks in tabular domains. For example, works on adversarial robustness in the context of decision tree-based classifiers often use per-feature constraints as adversarial constraints [5, 34, 35]. At the low level, these constraints are formalized either as bounds on L_∞ distance [5, 34], or using functions determining constraints for each feature value [35]. In these approaches, the feature constraints are independent. Such independence simplifies the problem. For example, the usage of L_∞ constraints enables to split a multidimensional optimization problem into a combination of simple one-dimension tasks [5], or to limit the set of points affected by the split change [35]. Unfortunately, per-feature constraints cannot realistically capture the *total cost* of mounting an attack: the aggregate cost of all the feature modifications required to produce an adversarial example, which is crucial to capture in tabular domains.

Also related to our cost-based proposal, Pierazzi et al. [142] introduce a general framework for defining attack constraints in the *problem space*. Our cost-based objective can be thought as an instance of this framework: we encode the problem-space constraints in the set of feasible examples.

Our cost model resembles the Gower distance [72], which is also a sum of "dissimilarities" across different categories of features. As opposed to this distance, our cost model can accommodate a wider class of numeric features, e.g., with a non-linear cost of changes. Also, it is not bounded to $[0, 1]$ interval providing flexibility to model a wider range of applications.

Utility-based objectives. The literature on *strategic classification* also considers utility-oriented objectives [50, 77, 126] for their agents. In this body of work, however, agents are not considered adversaries, and the gain is typically limited to {$+1$,-1}

reflecting the classifier decision. Our model supports arbitrary gain values, which enables us to model broader interests of the adversary such as revenue. Only the work by Sundaram et al. [167] supports gains different from $+1$ or -1, but they focus on PAC-learning guarantees in the case of linear classifiers, whereas our goal is to provide practical attack and defense algorithms for a wider family of classifiers.

5.6.2 Attack Strategies

Tabular domains. Several works have proposed attacks on tabular data. Ballet et al. [9] and Cartella et al. [30] propose to apply existing continuous attacks to tabular datasets. The authors focus on crafting imperceptible adversarial examples using standard methods from the image domain. They adapt these methods such that less "important" features (low correlation with the target variable) can be perturbed to a higher degree than other features. This corresponds to a special case within our framework, in which the feature-modification costs depend on the feature importance, with the difference that these approaches cannot guarantee that the proposed example will be feasible. Mathov et al. [125] propose to construct a surrogate model capable of mimicking the target classifier. A part of this surrogate model is a feature-embedding function which maps tabular data points to a homogeneous continuous domain. They apply projected gradient descent to produce adversarial examples in the embedding space and map the resulting examples back to the tabular domain. As opposed to our methods, Levy et al. cannot provide any guarantee that the produced adversarial example lay in the feasible set. Finally, Kantchelian et al. [93] propose a MILP-based attack and its relaxation within different L_p cost models against random-forest models. Our attack differs from these three methods as they use L_p or similar bounds, whereas we use a cost bound that can capture realistic constraints as explained in Sections 5.1 and 5.3.

Text domains. Our universal greedy attack algorithm is similar to the methods for attacking classifiers that operate on text [60, 111, 115, 179, 193, 200]. All these works, however, make use of adversarial constraints such as restrictions on the number of modified words or sentences. These constraints do not apply to tabular domains, as simply considering "number of changes" does not address the heterogeneity of features. Our algorithms also differ from these approaches in that we incorporate complex adversarial costs in the design of the algorithms. For example, the Greedy attack by Yang et al. [193] uses the target classifier's confidence for choosing the best modifications to create adversarial examples while accounting for the number of modifications. Our framework not only considers the number of modifications but also their cost, thus capturing richer constraints of the adversary.

5.7 Conclusions and Future Work

In this chapter, we have revisited the problem of adversarial robustness when the target machine-learning model operates on tabular data. We showed that previous approaches, tailored to produce adversarial image or text examples, and defend from them, perform poorly when used in tabular domains. This is because they are conceived within a threat model that does not capture the capabilities and goals of the tabular adversaries. We introduced a new framework to design attacks that account for the constraints existing in tabular adversarial scenarios: adversaries are limited by a budget to modify features, and can assign different utility to different examples. Having evaluated these attacks on three realistic datasets, we show that they effectively bring utility to the adversaries, which enables more realistic security evaluations of ML models. Further research is needed to study improvements to the attack algorithms, and to identify concrete methodologies for defining the cost and utility-based adversarial models.

Conclusions

In this thesis, we have re-evaluated the standard assumptions and approaches for mitigating and measuring privacy, security, and reliability risks in ML. Next, we provide a summary and key takeaways from each chapter.

In Chapter 2, we demonstrated that differentially private training guarantees consistent behavior between training and test time. By leveraging a notion of distributional generalization, we introduced new conceptual tools for designing deep learning methods, enabling us to mitigate unwanted behaviors and construct algorithms that outperform state-of-the-art approaches in distributional-robustness applications. The work introduced in the chapter showed that advances in the area of differential privacy can bring about improvements beyond just privacy.

At the same time, Chapter 3 highlighted the predictive-multiplicity cost in differentially private learning. We revealed that the randomization techniques used to ensure DP during model training can lead to significant variations in predictions for the same input example across equally-private models. The increase in predictive multiplicity with the level of privacy raises concerns about the justifiability of decisions supported by differentially private models in high-stakes settings. We emphasize the need for practitioners to audit the predictive multiplicity of DP-ensuring algorithms before deploying them in applications with individual-level consequences. Future work should focus on developing techniques to minimize or communicate the predictive multiplicity cost while preserving privacy guarantees.

In Chapter 4, we examined the phenomenon of disparate vulnerability in membership inference attacks (MIA). First, we established necessary and sufficient conditions for preventing MIAs, leveraging the concept of distributional generalization. These connections showed that a standard measure of vulnerability to membership inference is equivalent to an extended notion of generalization, implying that an improvement in either privacy or generalization necessarily implies an improvement in the other. We showed that accurately estimating disparate vulnerability requires careful consideration of suitable attack methods and the development of reliable statistical frameworks. Further research should explore effective mechanisms for preventing disparate vulnerability while preserving privacy.

In Chapter 5, we addressed the unique challenges of adversarial robustness in tabular domains, which are prevalent in safety-critical applications. We showed that existing threat models designed for image and text domains fail to account for the cost and utility considerations specific to tabular data. We proposed and evaluated cost and utility-aware threat models tailored to the capabilities and constraints of attackers targeting tabular domains. Future research should explore algorithmic improvements and systematic methodologies for defining useful cost and utility-aware constraints.

Broader Impact. We conclude with a discussion of the broader impact of the work.

Risk intersectionality. A common theme we have explored is the importance of considering diverse subpopulations in risk assessments. Although most research on algorithmic fairness focuses—for good reasons—on disparities in performance or outputs of the models, we demonstrate in Chapters 3 and 4 that bias also manifests in other model properties such as privacy and decision arbitrariness. By recognizing the potential disparities and inequities that can arise from inadequate privacy protections, our work promotes a more inclusive and fair approach to evaluating and mitigating diverse risks in ML.

Accounting for side effects of privacy. In Chapters 2 and 3, we have highlighted fundamental trade-offs in DP training. Although such training preserves privacy and causes predictable model behavior, it makes predictions of some inputs largely or fully explained by randomness used in training, thus arbitrary. Our work has provided a theoretical and practical framework for evaluating the exact level of decision arbitrariness for use by practitioners and regulators.

Improving security measures. Our work in Chapters 4 and 5 emphasizes the need for a broader view on the threat models and security measures in ML. In particular, by challenging the assumptions made in existing attack methodologies and advocating for practical constraints faced by adversaries, the research presented in Chapter 5 aims to enhance the effectiveness and real-world applicability of security or security-related measures. First, this can lead to more robust ML systems that are better equipped to detect and mitigate adversarial threats. Second, the proposed techniques are useful beyond security in settings where changing inputs to achieve a desired prediction is a legitimate means of achieving algorithmic recourse [170], contesting harmful systems [3, 105], or providing counterfactual explanations [175].

Informing policy and regulation. The insights and recommendations this thesis provides can inform the development of policies and regulations related to privacy, security, and reliability in ML. In Chapters 3 to 5, we highlight the unexpected effects of DP training on the arbitrariness of decisions, the importance of considering diverse subpopulations in risk measurements, and the need for realistic threat models. These findings can guide policymakers and regulatory bodies in formulating guidelines and standards that address the ethical and societal implications of ML technologies.

Appendices

Appendix A

Omitted Proofs

A.1 Proofs for Chapter 2

A.1.1 TV Stability implies Distributional Generalization

Proof of Theorem 2.1. First, observe that the following distributions are equivalent as the dataset is an i.i.d. sample:

$$\begin{aligned}
\Pr_{\substack{S\sim P^n\\ z\sim S}}[\phi(z;T(S))] &\equiv \Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z;T(S\cup\{z\}))],\\
\Pr_{\substack{S\sim P^n\\ z\sim P}}[\phi(z;T(S))] &\equiv \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z';T(S\cup\{z\}))].
\end{aligned} \tag{A.1}$$

It is thus sufficient to analyze the equivalent distributions instead. By the post-processing property of differential privacy, for any dataset $S \in \mathbb{D}^{n-1}$, any two examples $z, z' \in \mathbb{D}$, and any set $V \subseteq [0,1]$:

$$\Pr[\phi(z;T(S\cup\{z\})) \in V] \leq \Pr[\phi(z;T(S\cup\{z'\})) \in V] + \delta,$$

as datasets $S \cup \{z\}$ and $S \cup \{z'\}$ are neighbouring. Taking the expectation of both sides over $z, z' \sim P$ and $S \sim P^{n-1}$, we get:

$$\begin{aligned}
\Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z;T(S\cup\{z\})) \in V] &\leq \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z;T(S\cup\{z'\})) \in V] + \delta\\
&= \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z',T(S\cup\{z\})) \in V] + \delta,
\end{aligned} \tag{A.2}$$

where the last equality is simply renaming of the variables for convenience. Note that analogously we also can obtain a symmetric bound:

$$\Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z', T(S\cup\{z\}))\in V] \le \Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z; T(S\cup\{z\}))\in V] + \delta, \tag{A.3}$$

The total variation between these two distributions is bounded:

$$\begin{aligned} & d_{\mathsf{TV}}\Big(\Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z; T(S\cup\{z\}))], \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z', T(S\cup\{z\}))]\Big) \\ &= \sup_{V\subseteq \mathsf{range}(\phi)} \Big| \Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z; T(S\cup\{z\}))\in V] - \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z', T(S\cup\{z\}))\in V]\Big| \le \delta, \end{aligned}$$

where the last inequality is by Eq. (A.3). Using the equivalences in Eq. (A.1) we can see that:

$$d_{\mathsf{TV}}\Big(\Pr_{\substack{S\sim P^{n}\\ z\sim S}}[\phi(z; T(S))], \Pr_{\substack{S\sim P^{n}\\ z\sim P}}[\phi(z; T(S))]\Big) = \Big| \mathop{\mathbb{E}}_{\substack{S\sim P^{n}\\ z\sim S}}[\phi(z; T(S)] - \mathop{\mathbb{E}}_{\substack{S\sim P^{n}\\ z\sim P}}[\phi(z; T(S)] \Big| \le \delta,$$

which is the sought result. □

A.1.2 Tight Bound on TV Stability from DP

To prove Proposition 2.2.1, we make use of the hypothesis-testing interpretation of DP [181]. Let us define the hypothesis-testing setup and the two types of errors in hypothesis testing. For any two probability distributions P and Q defined over $\mathbb{D}$, let $\phi : \mathbb{D} \to \{0, 1\}$ be a *hypothesis-testing decision rule* that aims to tell whether a given observation from the domain $\mathbb{D}$ comes from P or Q.

Definition A.1 (Hypothesis-testing FPR and FNR)**.** Without loss of generality, the *false-positive error rate* α_ϕ (FPR, or type I error rate), and the *false-negative error rate* β_ϕ (FNR, or type II error rate) of the decision rule $\phi : \mathbb{D} \to [0, 1]$ are defined as the following probabilities:

$$\begin{aligned} \alpha_\phi &\triangleq \Pr_{z\sim P}[\phi(z) = 1] = \mathop{\mathbb{E}}_{P}[\phi], \\ \beta_\phi &\triangleq \Pr_{z\sim Q}[\phi(z) = 0] = 1 - \mathop{\mathbb{E}}_{Q}[\phi]. \end{aligned} \tag{A.4}$$

A well-known result due to Le Cam provides the following relationship between the trade-off between the two types of errors and the total variation between the probability distributions:

$$\alpha_\phi + \beta_\phi \geq 1 - d_{\mathsf{TV}}(P, Q). \tag{A.5}$$

DP is known to provide the following relationship between FPR and FNR of any decision rule:

Proposition A.1.1 (Kairouz et al. [92]). Suppose that an algorithm $T(S)$ satisfies (ϵ, δ)-DP. Then, for any decision rule $\phi : \mathbb{D} \rightarrow [0, 1]$:

$$\begin{aligned} \alpha_\phi + \exp(\epsilon)\, \beta_\phi \geq 1 - \delta, \\ \exp(\epsilon)\, \alpha_\phi + \beta_\phi \geq 1 - \delta. \end{aligned} \tag{A.6}$$

We can now prove Proposition 2.2.1:

Proof of Proposition 2.2.1. Consider a hypothesis-testing setup in which we want to distinguish between the distributions $T(S)$ and $T(S')$. Let us sum the two bounds in Eq. (A.6):

$$(\exp(\epsilon) + 1)(\alpha_\phi + \beta_\phi) \geq 2(1 - \delta) \implies \alpha_\phi + \beta_\phi \geq \frac{2 - 2\delta}{\exp(\epsilon) + 1}. \tag{A.7}$$

Let us take the optimal decision rule ϕ^*. In this case, the bound in Eq. (A.5) holds exactly:

$$d_{\mathsf{TV}}(T(S), T(S')) = 1 - (\alpha_{\phi^*} + \beta_{\phi^*}).$$

Combining this with Eq. (A.7), we get:

$$d_{\mathsf{TV}}(T(S), T(S')) \leq 1 - \frac{2 - 2\delta}{\exp(\epsilon) + 1} = \frac{\exp(\epsilon) - 1 + 2\delta}{\exp(\epsilon) + 1}.$$

□

Next, we show that the upper bound is tight:

Proposition A.1.2. There is an algorithm $T(S)$ satisfying (ε, δ)-DP, such that $d_{\mathsf{TV}}(T(S), T(S')) = \frac{\exp(\varepsilon) - 1 + 2\delta}{\exp(\varepsilon) + 1}$ for any two neighbouring datasets S and S'.

Proof. We use the construction of the reduced mechanism by Kairouz et al. [92]. Consider a mechanism $T : \{0, 1\} \rightarrow \{0, 1, 2, 3\}$, defined as follows:

$$\begin{aligned} &P(T(0) = 0) = 0 && P(T(1) = 0) = \delta \\ &P(T(0) = 1) = (1 - \delta) \cdot \tfrac{\exp(\epsilon)}{\exp(\epsilon) + 1} && P(T(1) = 1) = (1 - \delta) \cdot \tfrac{1}{\exp(\epsilon) + 1} \\ &P(T(0) = 2) = (1 - \delta) \cdot \tfrac{1}{\exp(\epsilon) + 1} && P(T(1) = 1) = (1 - \delta) \cdot \tfrac{\exp(\epsilon)}{\exp(\epsilon) + 1} \\ &P(T(0) = 3) = \delta && P(T(1) = 0) = 0 \end{aligned}$$

Observe that this mechanism satisfies (ε, δ)-DP, and $d_{\mathsf{TV}}(T(0), T(1)) = \frac{\exp(\varepsilon)-1+2\delta}{\exp(\varepsilon)+1}$. $\square$

A.1.3 Privacy Analysis of DP-IS-SGD

First, we present a loose analysis of the privacy guarantees of non-uniform Poisson subsampling.

Lemma A.1.1. Suppose that $T(S)$ satisfies (ϵ, δ)-DP and $\mathsf{Pois}(S)$ is a Poisson sampling procedure where each of the sampling probabilities p_i depend on the element z_i (but do not depend on the set S otherwise) and is guaranteed to satisfy $p_i \leq p^*$. Then $T \circ \mathsf{Pois}$ satisfies $(\ln(1 - p^* + p^* e^{\epsilon}), p^*\delta)$-DP. For small ϵ this can be bounded by $(\mathcal{O}(p^*\epsilon), p^*\delta)$-DP.

Proof of Lemma A.1.1. Consider two neighboring datasets S and $S' = S \cup \{z_0\}$ for some $z_0 \notin S$. We wish to show that for any set K, we have

$$\Pr(T(\mathsf{Pois}(S')) \in V) \leq (1 - p + pe^{\epsilon}) \Pr(T(\mathsf{Pois}(S)) \in V) + p\delta$$

and symmetrically for S and S'. We will only prove first of those inequalities, as the second is analogous.

Note that with probability $p_0 \leq p$ the element z_0 is included in $\mathsf{Pois}(S')$ and we have $\mathsf{Pois}(S') = \{z_0\} \cup \mathsf{Pois}(S)$, otherwise the element z_0 is not included, and conditioned on z_0 not being included $\mathsf{Pois}(S')$ has the same distribution as $\mathsf{Pois}(S)$. Therefore,

$$\Pr(T(\mathsf{Pois}(S')) \in V) = p_0 \Pr(T(\{z_0\} \cup \mathsf{Pois}(S)) \in V) + (1 - p_0) \Pr(T(\mathsf{Pois}(S)) \in V). \tag{A.8}$$

Now for each realization $\mathsf{Pois}(S) = \tilde{S}$, we have $\Pr(T(\{z_0\} \cup \tilde{S}) \in V) \leq e^{\epsilon} \Pr(T(\tilde{S}) \in V) + \delta$ by the assumed DP guarantee of the algorithm $T(S)$. We can average over all possible subsets $\tilde{S}$ to get

$$\begin{aligned}
\Pr(T(\{z_0\} \cup \mathsf{Pois}(S)) \in V) &= \sum_{\tilde{S}} \Pr(\mathsf{Pois}(S) = \tilde{S}) \Pr(T(\{z_0\} \cup \tilde{S}) \in V) \\
&\leq \sum_{\tilde{S}} \Pr(\mathsf{Pois}(S) = \tilde{S})(e^{\epsilon} \Pr(T(\tilde{S}) \in V) + \delta) \\
&= e^{\epsilon} \Pr(T(\mathsf{Pois}(S)) \in V) + \delta.
\end{aligned}$$

Plugging this back to the inequality (A.8), we get

$$\begin{aligned}
\Pr(T(\mathsf{Pois}(S')) \in V) &\leq p_0(e^{\epsilon} \Pr(T(\mathsf{Pois}(S)) \in V) + \delta) + (1 - p_0) \Pr(T(\mathsf{Pois}(S)) \in V) \\
&\leq (1 - p^* + p^* e^{\epsilon}) \Pr(T(\mathsf{Pois}(S)) \in V) + p^*\delta.
\end{aligned}$$

Finally, when $\epsilon \leq 1$ we have $e^{\epsilon} \leq (1 + 2\epsilon)$, and therefore $(1 - p^* + p^* e^{\epsilon}) \leq 1 + 2\epsilon p^* \leq$

$e^{2\epsilon p^*}$. □

For the tight privacy analysis of non-uniform Poisson subsampling, we make use of the notion of f-privacy:

Definition A.2 (f-Privacy Dong et al. [51])**.** An algorithm $T(S)$ satisfies f-privacy if for any two neighbouring datasets S, S' the following holds:

$$\tau(T(S), T(S')) \geq f,$$

where $\tau(P, Q)$ is a trade-off function between the FPR and FNR of distinguishing tests (see Appendix A.1.2):

$$\tau(P, Q)(\alpha) = \inf_{\phi\colon \mathbb{D} \to [0,1]} \{\beta_\phi : \alpha_\phi \leq \alpha\}, \tag{A.9}$$

and $f(\alpha) \in [0, 1]$ is a convex, continuous, non-increasing function.

Bu et al. [23] show that uniform Poisson subsampling (see Section 2.4.1) provides the following privacy amplification:

Proposition A.1.3 (Bu et al. [23])**.** Suppose that $T(S)$ satisfies f-privacy, and $\mathsf{Pois}(S)$ is a uniform Poisson sampling procedure with sampling probability $\bar{p}$. The composition $T \circ \mathsf{Pois}(S)$ satisfies f'-privacy with $f' = \bar{p}f + (1 - \bar{p})\mathsf{Id}$, where $\mathsf{Id}(\alpha) = 1 - \alpha$ is the trade-off function that corresponds to perfect privacy.

We show that a similar result holds for non-uniform Poisson subsampling:

Lemma A.1.2. Suppose that $T(S)$ satisfies f-privacy, and $\mathsf{Pois}(S)$ is a non-uniform Poisson sampling procedure, where the sampling probabilities p_i depend on the element z_i (but do not depend on the set S otherwise) and each is guaranteed to satisfy $p_i \leq p^*$. The composition $T \circ \mathsf{Pois}(S)$ satisfies f'-privacy with $f' = p^* + (1 - p^*)\mathsf{Id}$.

To show this, we adapt the proof Proposition A.1.3, and make use of the following lemma:

Lemma A.1.3 (Bu et al. [23])**.** Let $\{P_i\}_{i \in I}$ and $\{Q_i\}_{i \in I}$ be two collections of probability distributions on the same sample space for some index set I. Let $(\lambda_i)_{i \in I} \in [0, 1]^{|I|}$ be a collection of numbers such that $\sum_{i \in I} \lambda_i = 1$. If $\tau(P_i, Q_i) \geq f$ for all $i \in I$, then for any $p \in [0, 1]$:

$$\tau\left(\sum_i \lambda_i \cdot P_i,\ \sum_i (1 - p) \cdot \lambda_i \cdot P_i + \sum_i p \cdot \lambda_i \cdot Q_i\right) \geq pf + (1 - p)\mathsf{Id}.$$

Proof of Lemma A.1.2. We can think of the result of the subsampling procedure as outputting a binary vector $\vec{b} = (b_1, \ldots, b_n) \in \{0, 1\}^n$, where each bit b_i indicates whether an example $z_i \in S$ was chosen in the subsample or not. We denote the resulting subsample as $S_{\vec{b}} \subseteq S$. By definition of Poisson subsampling, each bit b_i is an independent sample $b_i \sim \mathsf{Bern}(p_i)$. Let us denote by $\lambda_{\vec{b}}$ the joint probability of $\vec{b}$. The composition $T(S) \circ \mathsf{Pois}(S)$ can be expressed as a mixture distribution:

$$T(S) \circ \mathsf{Pois}(S) = \sum_{\vec{b} \in \{0,1\}^n} \lambda_{\vec{b}} \cdot T(S).$$

Analogously, for a neighbouring dataset $S' \triangleq S \cup \{z_0\}$, with the sampling probability p_0 corresponding to z_0, we have:

$$T(S) \circ \mathsf{Pois}(S) = \sum_{\vec{b} \in \{0,1\}^n} p_0 \cdot \lambda_{\vec{b}} \cdot T(S'_{\vec{b}} \cup \{z_0\}) + \sum_{\vec{b} \in \{0,1\}^n} (1 - p_0) \cdot \lambda_{\vec{b}} \cdot T(S_{\vec{b}}).$$

Applying Lemma A.1.3, we get f_0-privacy with $f_0 = p_0 f + (1 - p_0)\mathsf{Id}$. Applying to an arbitrary other $z_0 \in \mathbb{D}$, we potentially get the worst-case privacy guarantee for the highest sampling probability, i.e., $f = p^* f + (1 - p^*)\mathsf{Id}$. □

Proposition 2.4.1 is immediate from Lemma A.1.2 by the fact that GDP is a special case of f-privacy.

A.1.4 Subgroup-level Distributional Generalization from TV Stability

TV-stability implies a more granular, subgroup-level notion of distributional generalization:

Definition A.3. Suppose that the data distribution P is a mixture of group-specific distributions P_G, for $G \in \mathbb{G}$. We define $(\delta, \mathbb{G})$-subgroup-DG similarly to δ-DG as follows:

$$\forall \phi, G \in \mathbb{G}: \quad \left| \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim S}} [\phi(z, T(S)) \mid z \in G, |S_G| > 0] - \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim P}} [\phi(z, T(S)) \mid z \in G] \right| \leq \delta,$$

where S_G denotes a subset of examples in the dataset S that belong to the group G.

Subgroup DG is a stronger notion of DG which says that the model's behavior on examples from each group in $\mathbb{G}$ distributionally generalizes in expectation, as long as the model encounters at least one representative of the group in training. In its definition, we explicitly prevent the case when the training dataset does not contain any group examples to avoid undefined behavior. Otherwise, it is unclear what is the group accuracy on the training dataset if there are no group representatives in the dataset.

We now show that TV stability implies this granular notion of DG:

Proposition A.1.4. δ-TV stability implies $(\delta, \mathbb{G})$-subgroup-DG for any group partitioning $\mathbb{G}$.

Proof. Observe that the following distributions are equivalent:

$$\begin{aligned} \Pr_{\substack{S\sim P^n\\ z\sim S}}[\phi(z;T(S)) \mid z \in G, |S_G| > 0] &\equiv \Pr_{\substack{S\sim P^{n-1}\\ z\sim P}}[\phi(z;T(S\cup\{z\}))], \\ \Pr_{\substack{S\sim P^n\\ z\sim P}}[\phi(z;T(S)) \mid z \in G] &\equiv \Pr_{\substack{S\sim P^{n-1}\\ z\sim P\\ z'\sim P}}[\phi(z';T(S\cup\{z\}))]. \end{aligned} \tag{A.10}$$

By applying each step from the proof of Theorem 2.1 to the equivalent distributions in Eq. (A.10), we have that the absolute gap between the distributions in Eq. (A.10) is bounded by δ. Concretely, the difference with the proof of Theorem 2.1 is that $z, z' \sim P_G$, not $z, z' \sim P$.

□

A.2 Proofs for Chapter 3

First, we provide an explanation on the range of disagreement without normalization:

Proposition A.2.1 (Range of non-normalized disagreement). The expression $\Pr[f_\theta(x) \neq f_{\theta'}(x)]$ has range of $[0, 0.5]$.

Proof. As $f_\theta(x) \in \{0,1\}$, we can assume $\Pr[f_\theta(x) = 1] = p$, and thus $\Pr[f_\theta(x) \neq f_{\theta'}(x)] = \Pr[f_\theta(x) = 0 \text{ and } f_{\theta'}(x) = 1] + \Pr[f_\theta(x) = 1 \text{ and } f_{\theta'}(x) = 0] = 2p(1-p) \in [0, 0.5]$. □

Next, we provide a proof that disagreement is proportional to variance in our setup:

Proof of Proposition 3.2.1. As $f_\theta(x) \in \{0,1\}$, we have that

$$\begin{aligned}
\mu(x) &= 2 \Pr_{\theta,\theta' \sim P_{T(S)}} [f_\theta(x) \neq f_{\theta'}(x)] \\
&= 2 \mathop{\mathbb{E}}_{\theta,\theta' \sim P_{T(S)}} [\mathbb{1}[f_\theta(x) \neq f_{\theta'}(x)]] \\
&= 2 \mathop{\mathbb{E}}_{\theta,\theta' \sim P_{T(S)}} [(f_\theta(x) - f_{\theta'}(x))^2] \\
&= 2 \mathop{\mathbb{E}}_{\theta \sim P_{T(S)}} [f_\theta^2(x)] - 4 \mathop{\mathbb{E}}_{\theta,\theta' \sim P_{T(S)}} [f_\theta(x) \cdot f_{\theta'}(x)] \\
&\quad + 2 \mathop{\mathbb{E}}_{\theta' \sim P_{T(S)}} [f_{\theta'}^2(x)] \qquad \text{(A.11)} \\
&= 4 \big(\mathop{\mathbb{E}}_{\theta \sim P_{T(S)}} [f_\theta(x)]^2 \\
&\quad - \mathop{\mathbb{E}}_{\theta \sim P_{T(S)}} [f_\theta(x)] \cdot \mathop{\mathbb{E}}_{\theta' \sim P_{T(S)}} [f_{\theta'}(x)] \big) \\
&= 4 \operatorname{Var}_{\theta \sim P_{T(S)}} (f_\theta(x)) \\
&= 4 p_x (1 - p_x),
\end{aligned}$$

where $p_x(1 - p_x)$ is the population variance of the r.v. $f_\theta(x) \sim \text{Bernoulli}(p_x)$. □

A.2.1 Closed-Form Characterization of Disagreement for Output Perturbation

Proof of Proposition 3.3.1. First, observe that the expression

$$p_x = \mathop{\mathbb{E}}_{\theta_{\mathsf{priv}} \sim P_{T(S)}} [f_{\theta_{\mathsf{priv}}}(x)]$$

can be expressed as:

$$\begin{aligned}
\mathbb{E}[f_{\theta_{\mathsf{priv}}}(x)] &= \mathbb{E}[\mathbb{1}[\mathsf{sigmoid}(\theta_{\mathsf{priv}}^\intercal x) > 0.5]] \\
&= \mathbb{E}[\mathbb{1}[\theta_{\mathsf{priv}}^\intercal x > 0]] \qquad \text{(A.12)} \\
&= \Pr(\theta_{\mathsf{priv}}^\intercal x > 0).
\end{aligned}$$

Denoting by $\xi \triangleq \mathcal{N}(0,1)$ and $\xi_d \triangleq \mathcal{N}(0, I_d)$, we can see that the score $\theta_{\mathsf{priv}}^{\mathsf{T}} x$ is equal to:

$$\begin{aligned}\theta_{\mathsf{priv}}^{\mathsf{T}} x &= (\theta_{\mathsf{np}} + \sigma \xi_d)^{\mathsf{T}} x \\ &= \theta_{\mathsf{np}}^{\mathsf{T}} x + \sigma \sum_{i=1}^{d} x_i \xi \\ &= \theta_{\mathsf{np}}^{\mathsf{T}} x + \sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sigma \xi \\ &= \theta_{\mathsf{np}}^{\mathsf{T}} x + \|x\| \sigma \xi.\end{aligned} \tag{A.13}$$

Plugging in the closed form in Eq. (A.13) into Eq. (A.12), we get:

$$p_x = \Pr(\theta_{\mathsf{np}}^{\mathsf{T}} x + \|x\| \sigma \xi > 0) = \Pr\left(\xi > -\frac{\theta_{\mathsf{np}}^{\mathsf{T}} x}{\|x\| \cdot \sigma}\right) = \Phi\left(\frac{\theta_{\mathsf{np}}^{\mathsf{T}} x}{\|x\| \cdot \sigma}\right). \tag{A.14}$$

□

A.2.2 Sample Complexity of Estimating Disagreement

Proof of Proposition 3.4.1. The $1/m-1$ term comes from Bessel's correction. Observe that

$$\begin{aligned}\mathbb{E}\left[\frac{m}{m-1} \hat{p}_x (1 - \hat{p}_x)\right] &= \frac{m}{m-1} (\mathbb{E}[\hat{p}_x] - \mathbb{E}[\hat{p}_x^2]) \\ &= \frac{m}{m-1} (\mathbb{E}[\hat{p}_x] - \mathrm{Var}(\hat{p}_x) - \mathbb{E}[\hat{p}_x]^2) \\ &= \frac{m}{m-1} \left(p_x - \frac{p_x(1-p_x)}{m} - p_x^2\right) \\ &= p_x(1-p_x)\end{aligned} \tag{A.15}$$

Therefore, $\mathbb{E}[\hat{\mu}(x)] = 4p_x(1-p_x) = \mu(x)$. □

Proof of Proposition 3.4.2. As $\hat{\mu}(x)$ is a continuous transformation of $\hat{p}_x$, we could bound the deviation $|\hat{\mu}(x) - \mu(x)|$ by $|\hat{p}_x - p_x|$. Suppose $\hat{p}_x = p_x + \nu$ and $\nu \in [-\eta, \eta]$, we have

$$\begin{aligned}
&\left|\frac{m}{m-1}\,\hat{p}_x(1-\hat{p}_x) - p_x(1-p_x)\right| = \\
&= \left|\frac{m}{m-1}\,(p_x+\nu)(1-p_x-\nu) - p_x(1-p_x)\right| \\
&= \left|\left(\frac{m}{m-1}-1\right)p_x(1-p_x) + \frac{m}{m-1}\nu(1-2p_x-\nu)\right| \\
&\leq \frac{p_x(1-p_x)}{m-1} + \frac{m}{m-1}|\nu||1-2p_x+\nu| \qquad \text{(A.16)} \\
&\leq \frac{p_x(1-p_x)}{m-1} + \frac{m}{m-1}|\nu|(1+|\nu|) \\
&\leq \frac{p_x(1-p_x)}{m-1} + \frac{m}{m-1}\eta(1+\eta) \\
&\leq \frac{1}{4(m-1)} + \frac{m}{m-1}\eta(1+\eta).
\end{aligned}$$

By Chernoff-Hoeffding inequality, we have the following concentration bounds on the sample mean $\hat{p}_x$,

$$\Pr[|\hat{p}_x - p_x| \geq \nu] \leq 2\exp\left(-2\nu^2 m\right). \qquad \text{(A.17)}$$

Thus with probability at least $1-\rho$, we have:

$$|\hat{p}_x - p_x| \leq \sqrt{\log(2/\rho)/2m}.$$

Combining Eq. (A.16) and Eq. (A.17), we have

$$\begin{aligned}
|\hat{\mu}(x) - \mu(x)| &= \left|\frac{4m}{m-1}\,\hat{p}_x(1-\hat{p}_x) - 4p_x(1-p_x)\right| \\
&\leq \frac{1}{(m-1)} + \frac{4m}{m-1}\eta(1+\eta).
\end{aligned} \qquad \text{(A.18)}$$

Plugging $\eta = \sqrt{\log(2/\rho)/2m}$ into Eq. (A.18) yields the desired result. Note that by solving $\frac{1}{(m-1)} + \frac{4m}{m-1}\eta(1+\eta) \leq \alpha$ with $\eta = \sqrt{\log(2/\rho)/2m}$ with conditions $\alpha > 0$ and $0 < \rho < 1$, we have:

$$m \geq 1 + \frac{\alpha + 2t(2+\alpha) + 2\sqrt{2}\sqrt{t(1+\alpha)(2t+\alpha)}}{\alpha^2}, \qquad \text{(A.19)}$$

where $t = \log(2/\rho)$. □

Proof of Proposition 3.4.3. Since the samples are i.i.d., we have the following union bound for the concentration of sample mean,

$$\begin{aligned}\Pr\left[\bigcup_{i=1}^{k}\{|\hat{p}_{x_i} - p_{x_i}| \geq \nu\}\right] &\leq \prod_{i=1}^{k}\Pr[|\hat{p}_{x_i} - p_{x_i}| \geq \nu] \\ &\leq 2k\exp\left(-2\nu^2 m\right).\end{aligned} \tag{A.20}$$

Therefore, with probability $1-\rho$, $|\hat{p}_{x_i} - p_{x_i}| \leq \sqrt{\log(2k/\rho)/2m}$ for $i = 1, \ldots, k$, and the desired result follows the derivation in Proposition 3.4.2.

□

A.3 Proofs for Chapter 4

A.3.1 Subgroup Vulnerability

To prove Proposition 4.3.2, we use two properties of a Bayes classifier. Within any subgroup of the data distribution, the subgroup-aware Bayes classifier performs optimally:

Lemma A.3.1. Consider random variables (X, T) where X takes values in $\mathbb{R}^d$, T takes values in a finite set $\mathbb{T}$, and a random variable Y that takes values in $\{0, 1\}$. Given a Bayes classifier $f^*_{X,T}(x, t)$, defined in this setting as the following minimizer:

$$f^*_{X,T}(x,t) \triangleq \arg\min_{f\colon\, \mathbb{R}^d \times \mathbb{T} \to \{0,1\}} \Pr[f(X,T) \neq Y], \tag{A.21}$$

we have for any $t \in \mathbb{T}$ and any other classifier $\theta \in \Theta$:

$$\Pr[f_\theta(X,T) \neq Y \mid T = t] \geq \Pr[f^*_{X,T}(X,T) \neq Y \mid T = t]. \tag{A.22}$$

Proof. We proceed by contradiction. Suppose that Eq. (A.22) does not hold, thus there exists $t' \in \mathbb{T}$ and a classifier $\theta' \in \Theta$ such that:

$$\Pr[f_{\theta'}(X,T) \neq Y \mid T = t'] < \Pr[f^*_{X,T}(X,T) \neq Y \mid T = t']. \tag{A.23}$$

But if that is the case, then we could construct a new classifier $f'(x, t)$ as follows:

$$f'(x,t) = \begin{cases} f_{\theta'}(x,t), & \text{if } t = t', \\ f^*_{X,T}(x,t), & \text{if } t \neq t'. \end{cases} \tag{A.24}$$

This classifier would attain lower expected loss than the optimal f^*, thus we have a contradiction. □

As a result, a Bayes classifier is a combination of optimal classifiers in each subgroup:

Lemma A.3.2. In the setting of Lemma A.3.1, suppose that $f^*_{X|T=t}(x)$ is a Bayes classifier for a subgroup:

$$f^*_{X|T=t}(x) \triangleq \arg \min_{f:\ \mathbb{R}^d \to \{0,1\}} \Pr[f(X) \neq Y \mid T = t] \tag{A.25}$$

Then, we have:

$$\Pr[f^*_{X,T}(X, T) \neq Y \mid T = t] = \Pr[f^*_{X|T=t}(X) \neq Y \mid T = t]. \tag{A.26}$$

Proof. As before, we proceed by contradiction.

If $\Pr[f^*_{X,T}(X, T) \neq Y \mid T = t] > \Pr[f^*_{X|T=t}(X) \neq Y \mid T = t]$, then $f^*_{X,T}$ is not a Bayes classifier as $f^*_{X|T=t}(X)$ relies on a post-processing of (X, T) and thus should be dominated by $f^*_{X,T}(x, t)$.

If $\Pr[f^*_{X,T}(X, T) \neq Y \mid T = t] < \Pr[f^*_{X|T=t}(X) \neq Y \mid T = t]$, then $f^*_{X|T=t}$ is not a conditional Bayes classifier. This is because any $f_{X,T} : \mathbb{R}^d \times \mathbb{T} \to \{0, 1\}$ can be seen as another classifier $f_X : \mathbb{R}^d \to \{0, 1\}$ as t is fixed, which should be dominated by $f^*_X(x)$.

We thus conclude that $\Pr[f^*_{X,T}(X, T) \neq Y \mid T = t] = \Pr[f^*_{X|T=t}(X) \neq Y \mid T = t]$. □

Next, we can prove Proposition 4.3.2:

Proof of Proposition 4.3.2. By Lemma A.3.1, for any adversary $\mathcal{A}_{\pi\circ g}(z; \theta)$ that uses features $\pi \circ g$ (see Section 4.2), the following holds:

$$\Pr[\mathcal{A}_{\pi\circ g}(z; \theta) = m \mid z \in G] \leq \Pr[\mathcal{A}^*_{\pi\circ g}(z; \theta) = m \mid z \in G] \tag{A.27}$$

over the randomness of the MIA game. This is the same as:

$$V_G(\mathcal{A}_{\pi\circ g}) \leq V_G(\mathcal{A}^*_{\pi\circ g}). \tag{A.28}$$

As $\mathcal{A}^*_\pi$ is an instance of $\mathcal{A}_{\pi\circ g}$, we have:

$$V_G(\mathcal{A}^*_\pi) \leq V_G(\mathcal{A}^*_{\pi\circ g}). \tag{A.29}$$

It remains to show that $V_G(\mathcal{A}^*_{\pi\circ g}) = \delta_G(\pi)$. By Lemma A.3.2:

$$V_G(\mathcal{A}^*_{\pi\circ g}) = \Pr[\mathcal{A}_{\pi\circ g}(z; \theta) \neq m \mid z \in G] = \Pr[\texttt{Att}^*_G(\pi(z; \theta)) \neq m \mid z \in G], \tag{A.30}$$

where $\mathtt{Att}^*_G(w)$ is the conditional Bayes classifier for group G:

$$\mathtt{Att}^*_G(w) \triangleq \arg\min_{\mathtt{Att}\colon \mathbb{W}\to\{0,1\}} \Pr[\mathtt{Att}(\pi(z;\theta)) \neq m \mid z \in G]. \tag{A.31}$$

Thus, by the relationship between the Bayes error and TV distance (Section 1.2), we have:

$$\Pr[\mathtt{Att}^*_G(\pi(z;\theta)) \neq m \mid z \in G] = d_{\mathsf{TV}}(P_{1,G}, P_{0,G}) = \delta_G(\pi). \tag{A.32}$$

□

A.3.2 Regular vs. Subgroup-Aware Vulnerability

Proof of Proposition 4.2.1. Observe that the features of the regular adversary $\pi(z;\theta)$ can be obtained from the features of the subgroup-aware adversary $(\pi(z;\theta), g(z))$. By the post-processing property of TV distance, we thus have that:

$$d_{\mathsf{TV}}(\pi\sharp P_1, \pi\sharp P_0) \leq d_{\mathsf{TV}}((\pi\circ g)\sharp P_1, (\pi\circ g)\sharp P_0) \tag{A.33}$$

By Proposition 4.3.1, this immediately implies $V(\mathcal{A}^*_\pi) \leq V(\mathcal{A}^*_{\pi\circ g})$, which is the sought inequality.

□

A.3.3 Bounds on Disparity from Algorithmic Fairness

Proof of Proposition 4.5.1. First, observe that a combination of the two conditions implies:

$$\delta_{G,G'}(\pi) = d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{0,G'}) \leq \eta + \nu$$

By this implication and the triangle property of total variation we have that:

$$\begin{aligned} d_{\mathsf{TV}}(\pi\sharp P_{0,G'}, \pi\sharp P_{1,G'}) &\leq \underline{d_{\mathsf{TV}}(\pi\sharp P_{1,G'}, \pi\sharp P_{0,G})} + d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{0,G'}) \\ &\leq \underline{d_{\mathsf{TV}}(\pi\sharp P_{1,G'}, \pi\sharp P_{0,G})} + \eta + \nu \end{aligned}$$

Applying the triangle inequality to the underlined term:

$$\begin{aligned} \underline{d_{\mathsf{TV}}(\pi\sharp P_{1,G'}, \pi\sharp P_{0,G})} &\leq d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{1,G}) + d_{\mathsf{TV}}(\pi\sharp P_{1,G}, \pi\sharp P_{1,G'}) \\ &\leq d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{1,G}) + \eta \end{aligned}$$

Combining the two,

$$\begin{aligned} d_{\mathsf{TV}}(\pi\sharp P_{0,G'}, \pi\sharp P_{1,G'}) - \eta - \nu &\leq \underline{d_{\mathsf{TV}}(\pi\sharp P_{1,G'}, \pi\sharp P_{0,G})} \\ &\leq d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{1,G}) + \eta \end{aligned}$$

Implying:

$$\delta_{G'}(\pi) - \delta_G(\pi) \leq 2\eta + \nu$$

If we apply the previous steps analogously we can also obtain:

$$\begin{aligned} d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{1,G}) - \eta - \nu &\leq d_{\mathsf{TV}}(\pi\sharp P_{1,G}, \pi\sharp P_{0,G'}) \\ &\leq d_{\mathsf{TV}}(\pi\sharp P_{0,G'}, \pi\sharp P_{1,G'}) + \eta \end{aligned}$$

Thus,

$$\delta_G(\pi) - \delta_{G'}(\pi) \leq 2\eta + \nu$$

Combining the inequalities, we get:

$$|\delta_G(\pi) - \delta_{G'}(\pi)| \leq 2\eta + \nu$$

By Corollary 4.3.3, we obtain the sought bound. □

A.3.4 Differential Privacy Bounds Subgroup Vulnerability and Disparity

Proof of Proposition 4.5.3. By Proposition A.1.4, we have that δ-TV stability implies:

$$d_{\mathsf{TV}}(P_{0,G}, P_{1,G}) \leq \delta. \tag{A.34}$$

By the post-processing property of TV distance, we have for any $\pi : \mathbb{D} \times \Theta \to \mathbb{R}^k$:

$$d_{\mathsf{TV}}(\pi\sharp P_{0,G}, \pi\sharp P_{1,G}) \leq \delta. \tag{A.35}$$

Thus, by Proposition 4.3.2, we have $V(\mathcal{A}^*_\pi) \leq \delta$. Moreover, by Corollary 4.3.2, we have $|\Delta V_{G,G'}(\mathcal{A}^*_\pi)| \leq \delta$. Applying the tight conversion from DP to TV stability in Proposition 2.2.1, we obtain the statement of the proposition.

□

Appendix B

Additional Discussion and Details for Chapter 2

B.1 Related Work Details

B.1.1 Differential Privacy and Robust Generalization

DP is known to imply a stronger notion of generalization, called *robust generalization*, which is a "tail bound" version of DG [13, 44, 57]. The original motivations for robust generalization are slightly different, but in our notation, a training procedure T is said to satisfy (γ, η)-Robust Generalization if and only if for any test $\phi : \mathbb{D} \times \Theta \to [0, 1]$, we have

$$\Pr_{S \sim \mathcal{D}^n}\left(| \mathop{\mathbb{E}}_{z \sim S} \phi(z; T(S)) - \mathop{\mathbb{E}}_{z \sim \mathcal{D}} \phi(z; T(S))| > \gamma \right) \leq \eta.$$

Any training method satisfying (ϵ, δ)-DP also satisfies $(\mathcal{O}(\epsilon), \mathcal{O}(\delta))$-robust generalization, as long as the sample size n is of size $\Omega(\log(1/\delta)/\epsilon^2)$ [13, Theorem 7.2], therefore it satisfies $\mathcal{O}(\epsilon + \delta)$-DG. Thus, it is possible to recover the result that DP implies DG as a consequence of these previous works, although with looser bounds.

The difference between Distributional Generalization and Robust Generalization is that DG considers all quantities in expectation, while robust generalization considers tail bounds with respect to the train dataset. We focus on DG for two reasons: First, we believe DG is conceptually simpler, as it can be seen as simply the TV distance between two natural distributions, and does not involve additional parameters. This simplicity is conceptually useful to the algorithm designer, but also enables us to prove simpler tight theoretical bounds that are independent of sample size. Second, it is often possible to lift results about DG to the stronger setting of robust generalization, with additional bookkeeping. Thus, we focus on DG in this chapter, with the understanding that stronger guarantees can be obtained for these methods if desired.

B.1.2 Information-Theoretic Generalization Bounds

Other than robust generalization, it is also possible to obtain bounds on generalization of arbitrary test (loss) functions using information-theoretic measures [149, 164, 191]. Recently, Steinke and Zakynthinou showed that one such information-theoretic measure—*conditional mutual information* (CMI) between training algorithm outputs and the training dataset—is bounded if the training algorithm is DP or TV stable. Thus, we could relate stability and DG as in Section 2.2.2 using CMI as an intermediate tool.

In particular, Steinke and Zakynthinou show that for a given $\phi : \mathbb{D} \times \Theta \to [0, 1]$:

$$\left| \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim S}} \phi(z; T(S)) - \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim P}} \phi(z; T(S)) \right| \leq \sqrt{\frac{2}{n} \cdot \mathrm{CMI}_P(T)}, \tag{B.1}$$

where $\mathrm{CMI}_{\mathrm{P}}(T)$ is the conditional mutual information of the training algorithm with respect to the data distribution. If the training algorithm satisfies $(\epsilon, 0)$-DP, they also show that $\mathrm{CMI}_{\mathrm{P}}(T) \leq \frac{n}{2}\epsilon^2$, where n is the dataset size. Plugging this into Eq. (B.1), we can see that the generalization upper bound (right-hand side) is ϵ. This is significantly looser than our bound in Section 2.2.2, as illustrated in Fig. 2.2.

A recent line of work on information-theoretic bounds explores sharper generalization bounds using individual-level measures [22, 76]. Analogously, as a direction for future work, it could also be possible to obtain tighter bounds on DG using per-instance notions of stability [63, 177].

B.1.3 Tension between Differential Privacy and Algorithmic Fairness

Beyond empirical observations that training with DP results in disparate impact on performance across subgroups [8, 145], Cummings et al. [45] and, more recently, Sanyal et al. [154] theoretically analyze the inherent tensions between DP and algorithmic fairness.

It might appear that this trade-off contradicts our results in which we claim that using DP or similar noise-adding algorithms with additional train-time interventions can reduce disparate impact. However, Cummings et al. [45] and Pujol et al. [145] discuss the relationship between privacy and disparate performance (accuracy or false-positive/false-negative rates), whereas we discuss the relationship between privacy and generalization. Even if a DP model has to incur at least a certain error on small subgroups on average [154, Lemma 1], this error is guaranteed to be similar at train time and test time (from our theoretical results in Section 2.2.2).

In terms of empirical results, the lower bound on subgroup error in Lemma 1 from Sanyal et al. [154] vanishes for subgroups of size greater than 100 even for small values of epsilon (e.g., 0.1). The subgroups and values of epsilon in our experiments in Section 2.5 are all larger than this, thus in our regime we can achieve meaningful subgroup performance using the DP-IS-SGD algorithm despite the fundamental trade-off.

B.2 Additional Details on Algorithms

We define q_G as the probability of group G, and m as the number of groups.

IS-SGD. The weight for group G is $w_G = {}^1\!/\!{}_{m \cdot q_g}$. Let g_i be the group that the i-th example belongs to. We then sample (with replacement) from the training set with the i-th example having a w_{g_i} chance of being sampled until we have b examples, where b is the batch size. Finally, for each mini-batch, we optimize the standard cross-entropy loss with the sampled examples.

IW-SGD. The weight for group G is $w_G = {}^1\!/\!{}_{m \cdot q_g}$. We optimize the following loss function:

$$w_g \cdot \ell(z; \theta),$$

where $\ell(z, \theta)$ is the cross-entropy loss and $z \in S$ drawn uniformly random drawn from the dataset, and G is the group to which z belongs.

B.3 Additional Experiment Details

B.3.1 Details on Datasets, Software, and Model Training

Table B.1: The number of examples in each subgroup for CelebA.

	training	validation	testing
not blond, female	71629	8535	9767
not blond, male	66874	8276	7535
blond, female	22880	2874	2480
blond, male	1387	182	180

Table B.2: The number of examples in each subgroup for UTKFace.

	training	validation	testing
male, White	3919	454	1105
male, Black	1700	181	437
male, Asian	1115	157	303
male, Indian	1594	190	477
male, Others	563	61	136
female, White	3316	384	902
female, Black	1606	188	414
female, Asian	1302	158	399
female, Indian	1230	152	333
female, Others	655	75	202

Table B.3: The number of examples in each subgroup for iNat.

	training	validation	testing
Actinopterygii	2112	195	312
Amphibia	14531	1242	1930
Animalia	5362	491	737
Arachnida	4838	461	660
Aves	191773	17497	26251
Chromista	435	52	55
Fungi	6148	575	883
Insecta	96894	8648	13013
Mammalia	26724	2475	3624
Mollusca	7627	693	1057
Plantae	159843	14653	22117
Protozoa	309	25	37
Reptilia	33404	2983	4494

Table B.4: The number of examples in each subgroup for CivilComments.

	training	validation	testing
Non-toxic, Identity	94895	15759	46185
Non-toxic, Other	143628	24366	72373
Toxic, Identity	18575	3088	9161
Toxic, Other	11940	1967	6063

Technical details. We use the following software:

- PyTorch [139] for implementing neural networks.
- opacus [195] for training PyTorch neural networks with DP-SGD.
- numpy [80], scipy [173], and pandas [135, 183] for numeric analyses.
- seaborn [180] for visualizations.

Table B.5: The number of examples in each subgroup for MNLI.

	training	validation	testing
Contradiction, No negation	57498	22814	34597
Contradiction, Negation	11158	4634	6655
Entailment, No negation	67376	26949	40496
Entailment, Negation	1521	613	886
Neutral, No negation	66630	26655	39930
Neutral, Negation	1992	797	1148

Table B.6: The number of examples in each subgroup for ADULT.

	training	validation	testing
Female, income$\leq$50k	11763	911	1749
Male, income$\leq$50k	18700	1373	2659
Female, income$>$50k	1444	105	220
Male, income$>$50k	8093	611	1214

- For gDRO [151], we use the implementation from the WILDS benchmark [99].

To train the models, we use Nvidia 2080ti, 3080, and A100 GPUs. Our experiments required approximately 400 hours of GPU time.

Datasets. For CelebA and CivilComments, we follow the training/validation/testing split in Koh et al. [99]. For UTKFace and iNat, we randomly split the data into 17000/2000/4708 and 550000/50000/75170 for training/validation/testing. For MNLI, we use the same training/validation/testing split in Sagawa et al. [151]. For Adult [100], we randomly split the data into 35000/3000/5842 for training/validation/testing. Tables B.1 to B.6 show the dataset statistics on each group.

All the datasets are publicly available for non-commercial use. In our work, we adhere to additional rules regulating the use of each dataset. All datasets other than iNat could potentially contain personally identifiable information, and are likely collected without consent, to the best of our knowledge. They are all, however, collected from manifestly public sources, such as public posts on social media. Thus, we consider the associated privacy risks low.

The data also contain offensive material (e.g., explicitly in the case of CivilComments dataset). We consider the associated risks of reproducing the offensive behavior low, as we use the datasets only to evaluate our theoretical and theoretically-inspired results.

Models. Similar to previous work [151], we use the ImageNet-1k pretrained ResNet50 [82] from torchvision for CelebA, UTKFace, and iNat, and use the pretrained BERT-Base [48] from huggingface [185] for CivilComments and MNLI.

For ADULT, we follow the setup in [192] and use logistic regression with standard optimization, and DP-based training methods. We fix the batch size to 256 (for SGD), weight decay to 0.01, and number of epochs to 20. For the DP algorithms, we use gradient norm clipping to 0.5, and sampling rate of 0.005. For all training algorithms, we train five model times with different random seeds and we record the mean and standard error of the mean of our metrics. The noise parameter σ for DP-SGD-F and DP-SGD is set to 1.0, and we set the σ for DP-IS-SGD to 5.0 to achieve similar privacy budget $\epsilon \approx 0.7$. The additional noise parameter for DP-SGD-F σ_2 is set to 10σ as in Xu et al. [192].

Hyperparameters. We run 50 epochs for CelebA, 100 epochs for UTKFace, 20 epochs for iNat, and 5 epochs for CivilComments and MNLI. For image datasets (CelebA, UTKFace, and iNat), we use the SGD optimizer and for NLP datasets (CivilComments and MNLI), we use the AdamW [120] optimizer. We use opacus's [195] implementation of DP-SGD and DP-AdamW to achieve DP guarantees.

We fix the batch size for none-DP algorithms to 64 for CelebA and UTKFace, 256 for iNat, 16 for CivilComments, and 32 for MNLI. For DP-SGD and DP-IS-SGD, we set the sample rate to 0.0001 for CelebA and iNat, 0.001 for UTKFace, and 0.00005 for CivilComments and MNLI.

B.3.2 Generalization of Worst-Case Group Accuracy as a Proxy for the DG Gap

Although generalization of worst-case group accuracy is not explicitly implied by DG, in our experiments it is practically equivalent to using the generalization gap of subgroup accuracy, which is bounded by TV stability. Let us first concretely define the generalization gap of the worst-case group accuracy:

Definition B.1. The on-average generalization gap of the worst-case accuracy is defined as the following difference:

$$\text{wggap} \triangleq \mathop{\mathbb{E}}_{S\sim P^n}\left[\max_{G\in\mathbb{G}} \mathop{\mathbb{E}}_{z\sim S_G}[\ell(z,\theta(S))] \,\middle|\, |S_G|>0\right] - \mathop{\mathbb{E}}_{S\sim P^n}\left[\max_{G\in\mathbb{G}} \mathop{\mathbb{E}}_{z\sim P_G}[\ell(z,\theta(S))]\right], \tag{B.2}$$

where we take $\ell((x,y),\theta) \triangleq \mathbb{1}[f_\theta(x) \neq y]$ to be the 0-1 loss. In this definition we explicitly restrict the datasets to include elements of each group $G \in \mathbb{G}$, which is a technicality needed in order to avoid undefined behavior.

In all our experimental results, the worst-performing groups (the maximizers in Eq. (B.2)) are always the same on the training and test data. As long as this holds—the worst-performing group is the same on the train and test data—the generalization gap above simplifies to:

$$\text{wggap} = \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim S_{G^*}}} [\ell(z, T(S)) \mid |S_{G^*}| > 0] - \mathop{\mathbb{E}}_{\substack{S \sim P^n \\ z \sim P_{G^*}}} [\ell(z, T(S))], \tag{B.3}$$

where $G^* \in \mathbb{G}$ is the worst-performing group. In Appendix A.1.4 we show that this simplified gap from Eq. (B.3) is bounded by TV stability.

Therefore, in practice the generalization gap in Eq. (B.2) offers a lower bound on the DG gap in Eq. (2.1). Using it as a proxy for DG gap follows the spirit of the estimation approach by Nakkiran and Bansal [129] which proposes to estimate the DG gap by taking the maximum of empirical generalization gaps for a finite set of relevant test functions (here, per-group accuracies).

Other Approaches to Estimate the DG Gap. The generalization gap of worst-case group accuracy can be loose as a proxy. Finding the worst-case test function is an object of study in the literature on *membership inference attacks* [161], because DG and the accuracy of such attacks in their standard formalization are equivalent, as showed in Section 4.3.1. In this work, we opt for a simpler and direct approach described above.

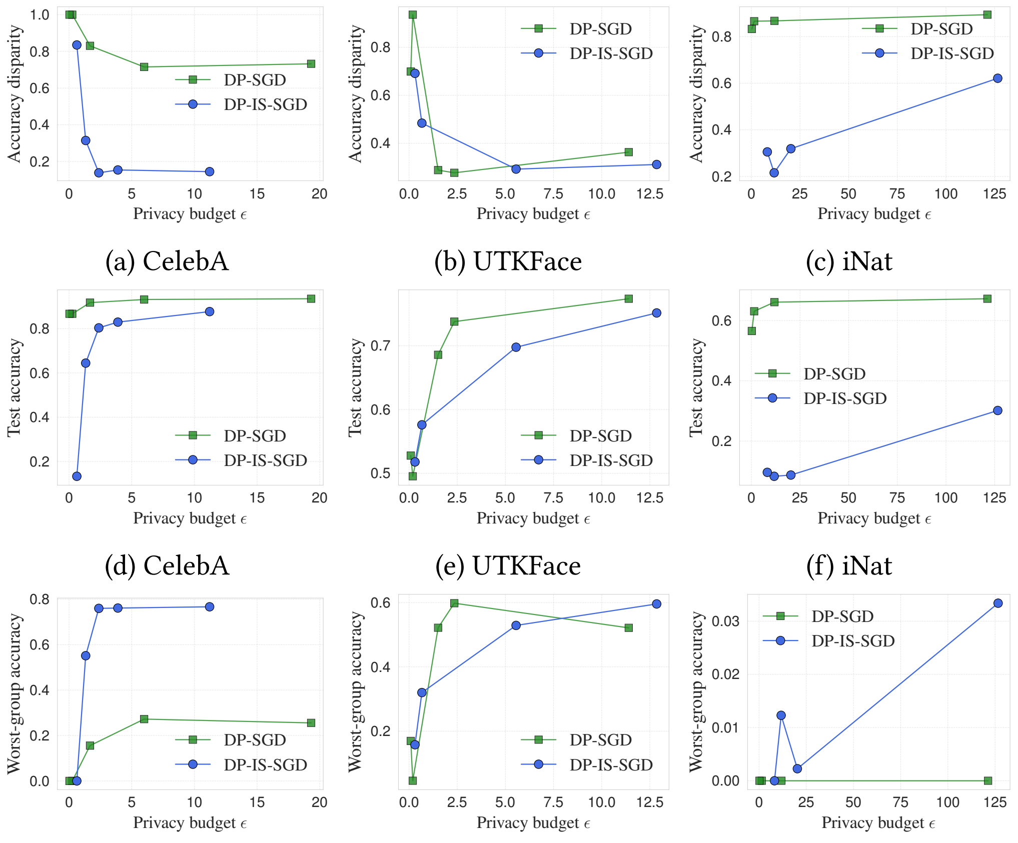
B.3.3 Additional Details for Section 2.5.2

Fig. B.1 shows the accuracy disparity, test accuracy, and worst-group accuracy for CelebA, UTKFace, and iNat on DP-SGD and DP-IS-SGD.

The reason that UTKFace has a similar disparity between DP-SGD and DP-IS-SGD is likely because UTKFace has a relatively small difference in the number of training examples between the largest group and the smallest group. In UTKFace, the majority group has around seven times more examples than in the minority group, whereas in CelebA, this difference is $52\times$.

Comparison with DP-SGD-F [192]. We did not manage to obtain good performance from DP-SGD-F on CelebA, UTKFace, and iNat, possibly because of the different domain—images—than tabular data considered by Xu et al. [192]. To proceed with the comparison, we evaluate the algorithms on the census data—ADULT dataset [100] (see Table B.6 for dataset statistics)—that Xu et al. [192] used in their work. As subgroups, we consider four intersectional groups composed of all possible values of the “sex” attribute and prediction class (an income higher/lower than 50k).

We show the results in Table B.7. For a comparable epsilon value (0.69 for DP-SGD-F, and 0.7 for our DP-IS-SGD), we see that our method has smaller accuracy disparity (Eq. 2) across the groups, although also lower overall accuracy.



(a) CelebA (b) UTKFace (c) iNat

(d) CelebA (e) UTKFace (f) iNat

(g) CelebA (h) UTKFace (i) iNat

Figure B.1: The disparity (lower the better) and test accuracies of the models trained with DP-SGD and IW-SGD on three datasets. If we care about privacy, DP-IS-SGD improves disparate impact at most privacy budgets. For CelebA, we train the model for 30 epochs. For UTKFace, we train for 100 epochs. For iNat, we train for 20 epochs. The GDP accountant is used to compute the privacy budget.

B.3.4 Additional Details for Section 2.5.3

We compare different algorithms, including SGD-L_2 and IW-SGD-L_2 as baselines, and two other algorithms, IS-SGD-L_2 [88] and gDRO-L_2 [151] in terms of the group robustness. We set the learning rate as 0.001 for CelebA, UTKFace, and iNat, 0.00002 for MNLI, and 0.00001 for CivilComments. We use the validation set to select the hyperparameters:

1. For SGD-L_2, IW-SGD-L_2, IS-SGD-L_2, and gDRO-L_2, we select the weight decay from 0.0001, 0.01, 0.1, and 1.0.
2. For DP-IS-SGD, we fix the gradient clipping to 1.0 (except for iNat, where we set the value to 10.0 as 1.0 does not converge). We select the noise parameter from

Table B.7: **DP-IS-SGD has lower disparity DP-SGD-F on ADULT and better accuracy at the same privacy level.** The table shows the privacy level, maximum accuracy disparity across groups, and overall accuracy for all algorithms.

Algorithm	ϵ	Accuracy disparity	Overall accuracy
SGD	–	0.660 ± 0.000	0.836 ± 0.000
DP-SGD	0.6573	0.852 ± 0.005	0.802 ± 0.001
DP-SGD-F	0.6964	0.657 ± 0.023	0.832 ± 0.001
DP-IS-SGD	0.7059	0.246 ± 0.034	0.766 ± 0.010

1.0, 0.1, 0.01, 0.001 on CelebA and UTKFace, select the noise parameter from 0.0000001, 0.000001, 0.00001, and 0.0001 on iNat and select the noise parameter from 0.01 and 0.001 on CivilComments and MNLI.

3. For IW-SGD-n, IS-SGD-n, and gDRO-n, we select the standard deviation of the random noise from 0.001, 0.01, 0.1, and 1.0 on CelebA, UTKFace, and iNat, and we select standard deviation of the random noise from 0.00001, 0.0001, and 0.001 on CivilComments and MNLI.

Statistical Concerns. Although our results appear to be comparable to or better than SOTA, we caution readers about the exact ordering of methods due to high estimation variance: these benchmarks have small validation and test sets (e.g., CelebA has 182 validation examples), and so hyperparameter tuning is subject to both overfitting and estimation error. For example, we observe validation accuracies which differ from their test accuracies by up to 5% in our experiments. We attempt to mitigate this using three random train/val/test splits on CelebA, and avoid large hyperparameter sweeps[1], but this is not done in prior work.

B.3.5 Additional Details for Section 2.5.4

We use the CIFAR-10 dataset [103], and ResNet-18 [82] as the network architecture. We train the model to be robust against L_∞ perturbations of at most $\varepsilon = 8/255$ bound, which is a standard setup for adversarial training on this dataset. We vary σ (noise parameter) from 0.0 (regular adversarial training without gradient noise) to 0.01. In addition, we compare the performance of noisy gradient to *adversarial training with early stopping*—a simple but effective approach for mitigating overfitting in adversarial training [147].

[1] For example, we do not tune the "group adjustments" parameter for gDRO, using the default from Koh et al. [99] instead.

In this experiment, we measure robust accuracy and its respective generalization gap, thus setting $\ell((x, y), \theta) \triangleq \mathbb{1}[f_\theta(x) \neq y]$ to be the 0-1 loss.

Appendix C

Additional Discussion and Details for Chapter 3

C.1 Multiplicity of Predictions vs. Scores

Recall that the models we consider are not only capable of outputting a binary prediction but also a confidence score. The disagreement metric in Eq. (3.1), however, only uses the predictions after applying a threshold. To verify if the trends we observe persist also at the level of confidence scores, we additionally evaluate *viable prediction range*, a metric for measuring multiplicity of the confidence scores proposed by Watson-Daniels et al. [182]:

$$\mu_{\mathsf{vp}}(x) \triangleq \max_{\theta \sim P_{T(S)}} h_\theta(x) - \min_{\theta \sim P_{T(S)}} h_\theta(x) \tag{C.1}$$

Fig. C.1 shows the viable prediction range for different values in the input space for logistic regression trained with objective perturbation on our synthetic dataset. The regions with high viable prediction range overlap with the regions with high disagreement (see Fig. 3.1). This is also consistent with the results on the tabular datasets, for which Fig. C.2 shows both disagreement and viable prediction range increasing on average as the level of privacy increases.

Implications. Models trained with a high level of privacy exhibit high multiplicity both of their confidence scores (in terms of viable prediction range) and of "hard" predictions after applying a threshold (in terms of disagreement).

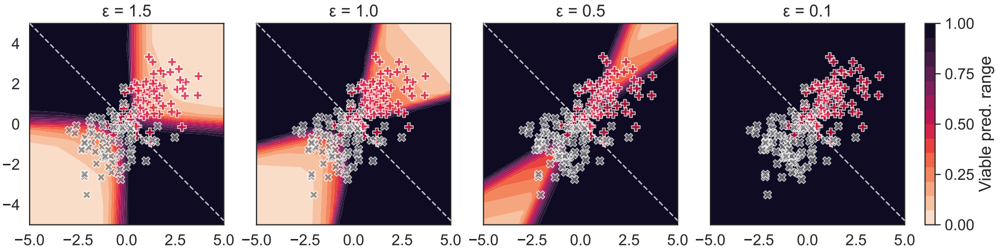


Figure C.1: Viable prediction range of logistic regression trained with objective perturbation is high for examples for which disagreement is also high. See Fig. 3.1 for the disagreement values and details of the plot setup.

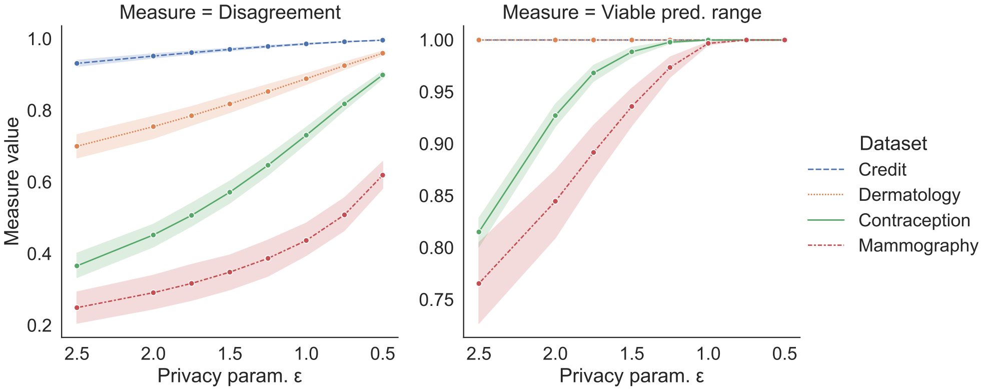


Figure C.2: Both the disagreement and viable prediction range of logistic regression trained with objective perturbation on tabular datasets increases as the level of privacy increases. See Fig. 3.3 for the details of the plot setup.

C.2 Experiment Details

C.2.1 Details on the Experiment Setup

Datasets. For illustrative purposes, we use the following classes as our target labels. For the Credit dataset, we use “Approved” as the target label. For the Contraception dataset, we use “long-term method”. For the dermatology dataset, we use “seboreic dermatitis” diagnosis. For the Mammography dataset, we use “malignant”.

CIFAR-10. We use the convolutional neural network trained over the ScatterNet features [134] following Tramer and Boneh [169, Table 9, Appendix]. We use DP-SGD with batch size of 2048, learning rate of 4, Nesterov momentum of 0.9, and gradient clipping norm of 0.1. We vary the gradient noise multiplier σ to achieve the privacy levels of $\epsilon \approx 2.22, 2.73, 3.62.4.39, 5.59$ as computed by the Moments accountant [1].

Software. We use the following software:

- diffprivlib [83] for the implementation of objective-perturbation for logistic regression.
- PyTorch [139] for implementing neural networks.
- opacus [195] for training PyTorch neural networks with DP-SGD.
- numpy [80], scipy [173], and pandas [135, 183] for numeric analyses.
- seaborn [180] for visualizations.

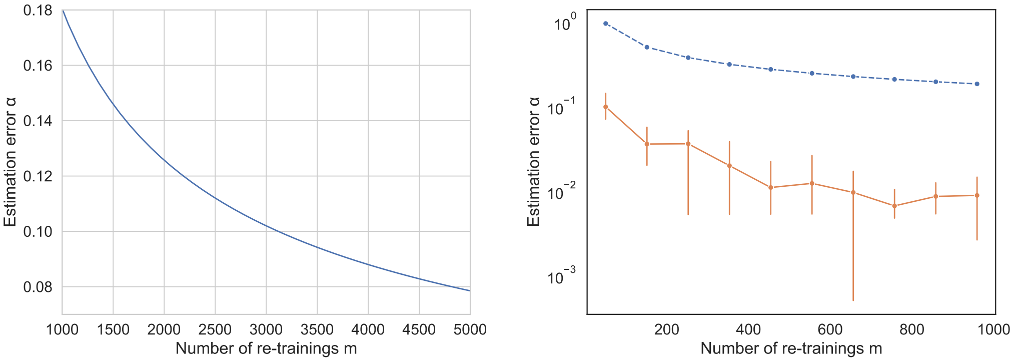
C.2.2 Additional Figures and Tables

The rest of the chapter contains additional figures and tables.

Table C.1: Summary statistics of the performance and predictive-multiplicity measures on real-world datasets. For tabular datasets, the performance metrics are the area under the ROC curve (AUC), and the harmonic mean of precision and recall (F_1 score) on the test data. For CIFAR-10, the performance metric is the accuracy on the test data. For these, we report mean and standard deviation over the m re-trained models. For disagreement, we report mean, standard deviation, minimum, median, maximum, the 90-th percentile, and the 95-th percentile over the examples in each respective test dataset. Observe that for every dataset there exist multiple examples with high level of predictive multiplicity even if the average level of predictive multiplicity for the given dataset is low. E.g., compare the 95-th percentile of disagreement on the CIFAR-10 dataset at $\epsilon = 2.22$ (0.81) to its mean value (0.11).

		AUC		F_1 score		Disagreement						
Dataset	ϵ	Mean	Std.	Mean	Std.	Mean	Std.	Min	Median	Max	90 pctl.	95 pctl.
Contraception	0.50	57.51	6.72	48.72	7.86	0.90	0.10	0.48	0.93	1.00	0.99	1.00
	0.75	60.26	6.20	50.29	7.54	0.82	0.17	0.24	0.88	1.00	0.99	1.00
	1.00	62.50	5.47	51.56	7.09	0.73	0.23	0.11	0.79	1.00	0.98	1.00
	1.25	64.27	4.71	52.62	6.62	0.65	0.27	0.05	0.70	1.00	0.97	0.99
	1.50	65.62	4.00	53.53	6.14	0.57	0.30	0.02	0.60	1.00	0.96	0.99
	1.75	66.65	3.38	54.31	5.67	0.51	0.32	0.00	0.50	1.00	0.95	0.99
	2.00	67.43	2.86	54.98	5.21	0.45	0.33	0.00	0.42	1.00	0.94	0.98
	2.50	68.49	2.10	55.97	4.39	0.37	0.33	0.00	0.27	1.00	0.92	0.97
Credit	0.50	52.22	15.95	46.48	16.38	1.00	0.00	0.99	1.00	1.00	1.00	1.00
	0.75	53.72	15.70	47.84	15.70	0.99	0.01	0.98	0.99	1.00	1.00	1.00
	1.00	55.16	15.41	49.15	15.05	0.99	0.01	0.96	0.99	1.00	1.00	1.00
	1.25	56.56	15.06	50.39	14.46	0.98	0.02	0.94	0.98	1.00	1.00	1.00
	1.50	57.86	14.69	51.59	13.89	0.97	0.03	0.91	0.98	1.00	1.00	1.00
	1.75	59.10	14.31	52.72	13.35	0.96	0.03	0.89	0.97	1.00	1.00	1.00
	2.00	60.26	13.91	53.77	12.85	0.95	0.04	0.86	0.96	1.00	1.00	1.00
	2.50	62.41	13.12	55.70	12.05	0.93	0.06	0.80	0.95	1.00	1.00	1.00
Dermatology	0.50	62.19	19.76	48.81	17.88	0.96	0.03	0.89	0.96	1.00	1.00	1.00
	0.75	66.75	17.65	52.67	16.44	0.93	0.05	0.79	0.93	1.00	0.99	0.99
	1.00	70.44	15.83	55.88	15.21	0.89	0.08	0.69	0.90	1.00	0.98	0.99
	1.25	73.46	14.28	58.57	14.20	0.85	0.10	0.60	0.86	1.00	0.98	0.98
	1.50	75.94	12.97	60.93	13.30	0.82	0.12	0.52	0.83	1.00	0.97	0.98
	1.75	78.04	11.89	62.98	12.60	0.79	0.13	0.46	0.80	1.00	0.95	0.97
	2.00	79.80	10.96	64.78	12.00	0.75	0.15	0.39	0.77	0.99	0.94	0.96
	2.50	82.66	9.45	67.80	10.95	0.70	0.17	0.32	0.72	0.99	0.92	0.94
Mammography	0.50	75.64	8.95	69.22	9.88	0.62	0.28	0.20	0.61	1.00	0.98	1.00
	0.75	78.57	6.51	72.46	7.04	0.51	0.34	0.07	0.45	1.00	0.98	1.00
	1.00	80.36	5.26	74.39	5.48	0.44	0.36	0.02	0.33	1.00	0.97	0.99
	1.25	81.62	4.44	75.64	4.66	0.39	0.37	0.01	0.24	1.00	0.95	0.99
	1.50	82.54	3.82	76.56	4.14	0.35	0.37	0.00	0.17	1.00	0.93	0.99
	1.75	83.25	3.36	77.29	3.81	0.32	0.36	0.00	0.12	1.00	0.91	0.98
	2.00	83.81	2.98	77.85	3.56	0.29	0.35	0.00	0.08	1.00	0.89	0.98
	2.50	84.61	2.40	78.70	3.22	0.25	0.34	0.00	0.04	1.00	0.84	0.96

		Accuracy		Avg. Disagreement across Classes						
Dataset	ϵ	Mean	Std.	Mean	Std.	Min	Median	Max	90 pctl.	95 pctl.
CIFAR-10	2.22	65.38	0.32	0.11	0.25	0.0	0.0	1.0	0.48	0.81
	2.73	67.65	0.35	0.09	0.23	0.0	0.0	1.0	0.36	0.77
	3.62	69.56	0.32	0.08	0.22	0.0	0.0	1.0	0.29	0.69
	4.39	70.38	0.33	0.07	0.21	0.0	0.0	1.0	0.23	0.64
	5.59	71.06	0.29	0.06	0.20	0.0	0.0	1.0	0.15	0.59



(a) Theoretical error of estimating disagreement w.p. 95%

(b) Empirical error of estimating disagreement for one arbitrarily chosen example (solid orange line —) compared to the theoretical maximum error w.p. 95% (dashed blue line – –). The error bars are 95% confidence intervals over 10 re-samplings of m models. This suggests that the theoretical upper bound on error is pessimistic in practice. y axis is logarithmic.

Figure C.3: Visualization of disagreement estimation error as a function of the number of models sampled from the training distribution $P_{T(S)}$.

Appendix D

Additional Details for Chapter 4

D.1 Additional Experiment Details

We use the following software:

- diffprivlib [83] for the implementation of objective-perturbation for logistic regression.
- fairlearn [16] for the implementation of algorithmic fairness post-processing.
- numpy [80], scipy [173], and pandas [135, 183] for numeric analyses.
- seaborn [180] for visualizations.

D.2 Additional Tables

The chapter contains additional tables.

Table D.1: Results of post-hoc tests on ADULT models. Columns: G *and* G'*:* identifiers of subgroups, t*:* value of the t statistic, p*:* uncorrected p-value, p*-corr.:* p-value after the correction for multiple comparisons.

NN-8	G	G'	t	p	p-corr.	NN-32	G'	G'	t	p	p-corr.
0	AE	AI	-4.4298	0.0000	**0.0001**	0	AE	AI	-11.3216	0.0000	**0.0000**
1	AE	BL	0.5143	0.6076	0.6751	1	AE	BL	0.9595	0.3385	0.3761
2	AE	OT	-1.7468	0.0822	0.1174	2	AE	OT	-4.1972	0.0000	**0.0001**
3	AE	WH	0.0498	0.9604	0.9604	3	AE	WH	0.5655	0.5724	0.5724
4	AI	BL	8.8677	0.0000	**0.0000**	4	AI	BL	24.1213	0.0000	**0.0000**
5	AI	OT	1.8976	0.0592	0.0987	5	AI	OT	6.1285	0.0000	**0.0000**
6	AI	WH	8.9236	0.0000	**0.0000**	6	AI	WH	25.4526	0.0000	**0.0000**
7	BL	OT	-2.6402	0.0089	0.0224	7	BL	OT	-6.4301	0.0000	**0.0000**
8	BL	WH	-1.3443	0.1804	0.2255	8	BL	WH	-1.2845	0.2005	0.2506
9	OT	WH	2.3290	0.0209	0.0417	9	OT	WH	6.1996	0.0000	**0.0000**

Table D.2: Results of post-hoc tests on Texas-50K models. See Table D.1 caption for details.

NN-32	G	G'	t	p	p-corr.	LR (Dem. Parity)	G'	G'	t	p	p-corr.
0	1	2	-3.4973	0.0006	**0.0007**	0	1	2	-1.2485	0.2133	0.3326
1	1	3	0.2056	0.8374	0.8374	1	1	3	-1.1910	0.2351	0.3326
2	1	4	4.2820	0.0000	**0.0000**	2	1	4	-2.4808	0.0139	0.0348
3	1	5	3.0576	0.0025	**0.0028**	3	1	5	-0.9385	0.3491	0.3879
4	2	3	10.0174	0.0000	**0.0000**	4	2	3	0.3151	0.7531	0.7531
5	2	4	21.2727	0.0000	**0.0000**	5	2	4	-3.4931	0.0006	**0.0020**
6	2	5	17.4069	0.0000	**0.0000**	6	2	5	1.1152	0.2661	0.3326
7	3	4	21.8804	0.0000	**0.0000**	7	3	4	-8.8594	0.0000	**0.0000**
8	3	5	13.2434	0.0000	**0.0000**	8	3	5	1.6787	0.0948	0.1896
9	4	5	-8.1600	0.0000	**0.0000**	9	4	5	12.8701	0.0000	**0.0000**

Table D.3: Results on ADULT, disaggregated by subgroups, for models with disparity F-test $p < 0.01$.

		Test acc.		Gen. gap		Subgroup vuln.	
		avg	std	avg	std	avg	std
Model	G						
32-Neuron NN	Amer-Indian-Eskimo	0.9028	0.0139	0.0115	0.0253	1.1701	4.8259
	Asian-Pac-Islander	0.8165	0.0119	0.0693	0.0195	5.7713	2.6300
	Black	0.9043	0.0049	0.0138	0.0086	0.8200	1.6261
	Other	0.8881	0.0179	0.0492	0.0295	3.2550	5.1807
	White	0.8338	0.0021	0.0109	0.0035	0.9773	0.4496
8-Neuron NN	Amer-Indian-Eskimo	0.9042	0.0151	0.0041	0.0281	0.3701	4.7177
	Asian-Pac-Islander	0.8264	0.0119	0.0223	0.0214	2.1320	2.7965
	Black	0.9066	0.0047	0.0035	0.0093	0.1878	1.6152
	Other	0.8913	0.0165	0.0149	0.0309	1.2805	5.6344
	White	0.8345	0.0020	0.0039	0.0036	0.3535	0.4314

Table D.4: Results on Texas-50K, disaggregated by subgroups, for models with disparity F-test $p < 0.01$.

		Test acc.		Gen. gap		Subgroup vuln.	
		avg	std	avg	std	avg	std
Model	G						
32-Neuron NN	1	0.8699	0.0380	0.0791	0.0451	8.5188	8.2829
	2	0.8644	0.0153	0.1013	0.0180	10.7429	3.0129
	3	0.8498	0.0085	0.0855	0.0106	8.3947	1.6121
	4	0.8644	0.0066	0.0637	0.0063	6.0331	0.8261
	5	0.8708	0.0063	0.0697	0.0074	6.7288	1.0840
Fair LR (Dem. Parity)	1	0.6932	0.0562	-0.0010	0.0839	0.0075	8.9200
	2	0.6934	0.0203	0.0095	0.0295	0.8381	2.9201
	3	0.7323	0.0084	0.0143	0.0099	0.7667	1.1361
	4	0.7771	0.0027	0.0155	0.0048	1.5751	0.4952
	5	0.7384	0.0068	0.0106	0.0088	0.5997	0.8448

Appendix E

Additional Discussion and Details for Chapter 5

E.1 Other Possible Adversarial Objectives

We propose a cost-oriented and a utility-oriented adversarial objective in Section 5.3. These are not the only possible formalizations for our high-level goals. One other approach is an adversary maximizing utility subject to a cost budget:

$$\begin{aligned} \max_{x \in \mathcal{F}(x,y)} & \mathbb{1}[f(x) \neq y] \cdot u_{x,y}(x') \\ & = \mathbb{1}[f(x) \neq y] \cdot [r(x') - c(x, x')]_+ \\ & \text{s.t. } c(x, x') < \gamma \end{aligned} \tag{E.1}$$

This formalization is a middle ground between our cost-constrained and utility-constrained objectives: On the one hand, the adversary is aware of the utility of a given example. On the other hand, they do not adjust their budget for different examples, i.e. the constraint for \$10 and \$1,000 stays the same, even though the adversary clearly differentiates in their value. We conducted preliminary experiments with this objective, and its results are marginally different from the cost-bounded one in our experimental setup.

E.2 Additional Details on the Experiments

We provide the details of our experimental setup.

Table E.1: IEEE-CIS and HomeCredit attack and defense parameters

Dataset	**Parameter**	**Value range**
IEEE-CIS	*Max. iterations*	100K
	γ *(for CB attacks)*	$[1, 3, 10, 30]$
	τ *(for UB attacks)*	$[0, 10, 50, 500, 1000]$
HomeCredit	*Num. of iterations*	100
	γ *(for CB attacks)*	$[1, 10, 100, 1K, 10K]$
	τ *(for UB attacks)*	$[10K, 300K, 400K, 500K, 600K, 800K]$

E.2.1 Software

We use the following software:

- PyTorch [139] for implementing neural networks.
- numpy [80], scipy [173], and pandas [135, 183] for numeric analyses.
- seaborn [180] for visualizations.

E.2.2 Hyperparameter Selection

We list our defense and attack parameters in Table E.1. The TabNet parameters are denoted according to the original paper [6]. We set the virtual batch size to 512. As training the clean baseline for HomeCredit was prone to overfitting in our setup, we reduced the number of training epochs to 100. Other hyperparameters were selected with a grid search.

E.2.3 Dataset Processing and Adversarial Cost Models

For each dataset, we create an adversarial cost based on hypothetical scenarios. In this section, we describe how we process the data, and how we assign costs to modifications of the features in each dataset.

TwitterBot. We use 19 numeric features from this dataset. We drop three features for which we cannot compute the effect of a transformation as we do not have access to the original tweets. We use the number of followers as the adversary's gain. We assign costs of features based on estimated costs to purchase Twitter accounts of different characteristics on darknet markets.

IEEE-CIS. We ascribe cost of changes, assuming that the adversary can change the device type and email address at a small cost. The device type can be changed with low effort using specific software. Email domain can be changed with a registration of a new email address which typically cannot be automated. Although also low cost, it takes more time and effort than changing the device time. We reflect these assumptions ascribing the costs \$0.1 and \$0.2 to these changes. Changing the type of the payment card requires obtaining a new card, which costs approximately \$20 in US-based darknet marketplaces as of the time of writing. We consider the transaction amount as a gain obtained by an adversary.

HomeCredit. The main goal of the adversary in this task is receiving a credit approval. As one example represents a loan application, we set the credit amount to be the gain of the example. All features which can be used by an adversary are listed in Table E.4 with the costs we ascribe to them. We assume six groups of features and estimate the cost as follows:

- *Group 1*: Features that an adversary can change with negligible effort such as an email address, weekday, or hour of the loan application. We ascribe \$0.1 cost to these transformations.
- *Group 2*: Features associated to income. We use these as numerical features to illustrate the flexibility of our method. We assume that to increase income by \$1, the adversary needs to pay \$1.
- *Group 3*: Features associated to changing a phone number. Based on the US darknet marketplace prices as of the time of writing, we estimate that purchasing a SIM card costs \$10.
- *Group 4*: Features related to official documents which can be temporally changed. For example, a car's ownership can be transferred from one person to another for the application period, and returned to the original owner after it. We ascribe a cost of \$100 to these changes.
- *Group 5*: Features that require either document forging or permanent changes to a person's status. For instance, purchasing a fake university diploma. For the sake of an example, we estimate their cost at \$1,000.
- *Group 6*: Features related to credit scores provided by external credit-scoring agencies. We estimate the cost of changes in this group with a manipulation model based on a real-world phenomenon of credit piggybacking, described next.

Credit-Score Manipulation. In our feature set we include the features that contain credit scores from unspecified external credit-scoring agencies. One reported way of

Table E.2: Costs of changing a feature on the TwitterBot dataset

Feature	**Estimated cost, $**
likes_per_tweet	0.025
retweets_per_tweet	0.025
user_tweeted	2
user_replied	2

Table E.3: Costs of changing a feature on the IEEE-CIS dataset

Feature	**Estimated cost, $**
DeviceType	0.1
P_emaildomain	0.2
card_type	20

Table E.4: Costs of changing a feature on the HomeCredit dataset

Feature	**Estimated cost, $**
name_contract_type	0.1
name_type_suite	0.1
flag_email	0.1
weekday_appr_process_start	0.1
hour_appr_process_start	0.1
amt_income_total	1
flag_emp_phone	10
flag_work_phone	10
flag_cont_mobile	10
flag_mobil	10
flag_own_car	100
flag_own_realty	100
reg_region_not_live_region	100
reg_region_not_work_region	100
live_region_not_work_region	100
reg_city_not_live_city	100
reg_city_not_work_city	100
live_city_not_work_city	100
name_income_type	100
cluster_days_employed	100
name_housing_type	100
occupation_type	100
organization_type	100
name_education_type	1000
name_family_status	1000
has_children	1000

affecting such credit scores is using credit piggybacking [1]. During piggybacking, a rating buyer finds a “donor” willing to share a credit for a certain fee. We introduce a model that captures costs of manipulating a credit score through piggybacking.

We assume that after one piggybacking manipulation the rating is averaged between “donor” and recipient, and that “donors” have the maximum rating (1.0). Then, the cost associated to increasing the rating from, e.g., 0.5 to 0.75 is the same as that of increasing from 0.9 to 0.95. This cost cannot be represented by a linear function. Let the initial score value be x. The updated credit score after piggybacking is $x' = {}^{(x+1)}/_{2}$. If we repeat the operation n times, the score becomes:

$$x' = \frac{x + 2^n - 1}{2^n}$$

Thus, the number of required piggybacking operations can be computed from the desired final score x' as $n = \log_2 \frac{1-x}{1-x'}$, and the total cost is $c(x, x') = nC$, where C is the cost of one operation. For the sake of an example, we estimate it to be \$10,000.

$$c(x, x') = C \log_2 \frac{1-x}{1-x'} = C(\log_2(1-x) - \log_2(1-x'))$$

This is not a fully realistic model, as we cannot know how exactly credit score agencies compute the rating. However, it is based on a real phenomenon and enables us to demonstrate our framework’s support of non-linear costs.

[1] https://www.experian.com/blogs/ask-experian/what-is-piggybacking-credit

Table E.5: Effect of beam size B in the Universal Greedy algorithm on the IEEE-CIS dataset. The success rates are close for all choices of the beam size, thus the beam size of one offers the best performance in terms of runtime.

	Adv. success, %			
Cost bound →	10	30	Gain	∞
Beam size ↓				
1	45.32	**56.57**	**56.22**	**68.20**
10	45.32	56.01	55.65	56.01
100	45.32	56.53	56.18	56.53

	Success/time ratio			
Cost bound →	10	30	Gain	∞
Beam size ↓				
1	**3.78**	**4.80**	**2.53**	**2.06**
10	2.14	2.25	1.31	1.15
100	0.66	0.65	0.65	0.66

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, 2016.

[2] Kendra Albert, Jonathon Penney, Bruce Schneier, and Ram Shankar Siva Kumar. Politics of adversarial machine learning. *arXiv preprint arXiv:2002.05648*, 2020.

[3] Kendra Albert, Maggie Delano, Bogdan Kulynych, and Ram Shankar Siva Kumar. Adversarial for good? how the adversarial ml community's values impede socially beneficial uses of attacks. *arXiv preprint arXiv:2107.10302*, 2021.

[4] Ali Alkhatib. To live in their utopia: Why algorithmic systems create absurd outcomes. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, 2021.

[5] Maksym Andriushchenko and Matthias Hein. Provably robust boosted decision stumps and trees against adversarial attacks. *Advances in Neural Information Processing Systems*, 2019.

[6] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *AAAI*, 2021.

[7] Peter Auer, Mark Herbster, and Manfred KK Warmuth. Exponentially many local minima for single neurons. *Advances in neural information processing systems*, 8, 1995.

[8] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

[9] Vincent Ballet, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, Marcin Detyniecki, et al. Imperceptible adversarial attacks on tabular data. In *Robust AI in FS NeurIPS Workshop*, 2019.

[10] Rina Foygel Barber and John C Duchi. Privacy and statistical risk: Formalisms and minimax bounds. *arXiv preprint arXiv:1412.4451*, 2014.

[11] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 2016.

[12] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Asia-CCS*, 2006.

[13] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan R. Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2016.

[14] Arindrajit Basu, Elonnai Hickok, and Aditya Singh Chawala. The Localisation Gambit: Unpacking Policy Measures for Sovereign Control of Data in India. *Centre for Internet and Society, India*, 2019.

[15] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 2018.

[16] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, May 2020. URL https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/.

[17] Emily Black, Klas Leino, and Matt Fredrikson. Selective ensembles for consistent predictions. In *International Conference on Learning Representations*, 2021.

[18] Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022.

[19] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 World Wide Web conference*, 2019.

[20] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2002.

[21] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3), 2001.

[22] Yuheng Bu, Shaofeng Zou, and Venugopal V Veeravalli. Tightening mutual information-based bounds on generalization error. *IEEE Journal on Selected Areas in Information Theory*, 2020.

[23] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020.

[24] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, 2018.

[25] Jonathon Byrd and Zachary Chase Lipton. What is the effect of importance weighting in deep learning? In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019.

[26] Stefano Calzavara, Claudio Lucchese, Gabriele Tolomei, Seyum Assefa Abebe, and Salvatore Orlando. Treant: training evasion-aware decision trees. *Data Min. Knowl. Discov.*, 2020.

[27] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[28] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017.

[29] Michele Carminati, Luca Santini, Mario Polino, and Stefano Zanero. Evasion attacks against banking fraud detection systems. In *RAID*, 2020.

[30] Francesco Cartella, Orlando Anunciacao, Yuki Funabiki, Daisuke Yamaguchi, Toru Akishita, and Olivier Elshocht. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *SafeAI Workshop at AAAI*, 2021.

[31] Hongyan Chang and Reza Shokri. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2021.

[32] Kamalika Chaudhuri and Nina Mishra. When random sampling preserves privacy. In *Annual International Cryptology Conference*. Springer, 2006.

[33] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[34] Hongge Chen, Huan Zhang, Duane Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. In *ICML*, 2019.

[35] Yizheng Chen, Shiqi Wang, Weifan Jiang, Asaf Cidon, and Suman Jana. Cost-aware robust tree ensembles for security applications. In *USENIX*, 2021.

[36] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. F-BLEAU: Fast black-box leakage estimation. In *IEEE Symposium on Security and Privacy, S&P*, 2019.

[37] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 2017.

[38] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[39] Danielle Keats Citron and Frank Pasquale. The scored society: Due process for automated predictions. *Wash. L. Rev.*, 89, 2014.

[40] Amanda Coston, Ashesh Rambachan, and Alexandra Chouldechova. Characterizing fairness over the set of good models under selective labels. In *International Conference on Machine Learning*. PMLR, 2021.

[41] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

[42] Kathleen Creel and Deborah Hellman. The algorithmic Leviathan: Arbitrariness, fairness, and opportunity in algorithmic decision-making systems. *Canadian Journal of Philosophy*, 2022.

[43] Francesco Croce and Matthias Hein. On the interplay of adversarial robustness and architecture components: patches, convolution and attention. *arXiv preprint arXiv:2209.06953*, 2022.

[44] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Proceedings of the 29th Conference on Learning Theory, COLT*, 2016.

[45] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 2019.

[46] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of A*. *J. ACM*, 1985.

[47] Ambra Demontis, Marco Melis, Battista Biggio, Davide Maiorca, Daniel Arp, Konrad Rieck, Igino Corona, Giorgio Giacinto, and Fabio Roli. Yes, machine learning can be more secure! a case study on android malware detection. *IEEE transactions on dependable and secure computing*, 2017.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[49] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[50] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *EC*, 2018.

[51] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

[52] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[53] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

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

[55] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012.

[56] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 2014.

[57] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015.

[58] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing, STOC*, 2015.

[59] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 2020.

[60] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *ACL*, 2018.

[61] Michael D. Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. Privacy for all: Ensuring fair and equitable privacy protections. In *Conference on Fairness, Accountability and Transparency, FAT*, 2018.

[62] Farhad Farokhi and Mohamed Ali Kaafar. Modelling and quantifying membership information leakage in machine learning. *arXiv preprint arXiv:2001.10648*, 2020.

[63] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a Renyi filter. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[64] 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), 2019.

[65] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. The (im) possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM*, 64(4), 2021.

[66] Georgi Ganev, Bristena Oprisanu, and Emiliano De Cristofaro. Robin hood and matthew effects: Differential privacy has disparate impact on synthetic data. In *International Conference on Machine Learning*. PMLR, 2022.

[67] Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. Optimal noise-adding mechanism in additive differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2019.

[68] Salah Ghamizi, Maxime Cordy, Martin Gubri, Mike Papadakis, Andrey Boytsov, Yves Le Traon, and Anne Goujon. Search-based adversarial testing and improvement of constrained credit scoring systems. In *ESEC/FSE*, 2020.

[69] Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. Classification of twitter accounts into automated agents and human users. In *ASONAM*, 2017.

[70] Karan Goel, Albert Gu, Yixuan Li, and Christopher Re. Model patching: Closing the subgroup performance gap with data augmentation. In *8th International Conference on Learning Representations, ICLR*, 2020.

[71] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[72] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 1971.

[73] Graham Greenleaf and Bertil Cottier. 2020 ends a decade of 62 new data privacy laws. *Privacy Laws & Business International Report*, 2020.

[74] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 2009.

[75] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[76] Mahdi Haghifam, Jeffrey Negrea, Ashish Khisti, Daniel M Roy, and Gintare Karolina Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[77] Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. In *ITCS*, 2016.

[78] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NIPS*, 2016.

[79] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016.

[80] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 2020.

[81] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Sys. Sci. and Cybernetics*, 1968.

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

[83] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the IBM differential privacy library. *arXiv preprint arXiv:1907.02444*, 2019.

[84] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018.

[85] Hsiang Hsu and Flavio du Pin Calmon. Rashomon capacity: A metric for predictive multiplicity in probabilistic classification. *Advances in Neural Information Processing Systems*, 2022.

[86] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

[87] Thomas Humphries, Matthew Rafuse, Lindsey Tulloch, Simon Oya, Ian Goldberg, Urs Hengartner, and Florian Kerschbaum. Differentially private learning does not bound membership inference. *arXiv preprint arXiv:2010.12112*, 2020.

[88] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, 2022.

[89] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.

[90] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2021.

[91] Jinyuan Jia and Neil Zhenqiang Gong. Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In *USENIX*, 2018.

[92] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

[93] Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and hardening of tree ensemble classifiers. In *ICML*, 2016.

[94] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *ACM Symposium on Theory of Computing*, 1994.

[95] Samir Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 1999.

[96] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *Uncertainty in Artificial Intelligence*. PMLR, 2022.

[97] Klim Kireev, Bogdan Kulynych, and Carmela Troncoso. Adversarial robustness for tabular data through cost and utility awareness. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium*, 2023.

[98] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 2020.

[99] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

[100] Ron Kohavi et al. Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid. In *KDD*, 1996.

[101] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. Adversarial malware binaries: Evading deep learning for malware detection in executables. In *2018 26th European signal processing conference (EUSIPCO)*. IEEE, 2018.

[102] Richard E. Korf. Iterative-Deepening-A*: An optimal admissible tree search. In *Joint Conference on Artificial Intelligence*, 1985.

[103] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[104] Bogdan Kulynych, Jamie Hayes, Nikita Samarin, and Carmela Troncoso. Evading classifiers in discrete domains with provable optimality guarantees. *arXiv preprint arXiv:1810.10939*, 2018.

[105] Bogdan Kulynych, Rebekah Overdorf, Carmela Troncoso, and Seda F. Gürses. POTs: protective optimization technologies. In *FAT**, 2020.

[106] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. Disparate vulnerability to membership inference attacks. *Proceedings on Privacy Enhancing Technologies*, 1, 2022.

[107] Bogdan Kulynych, Yao-Yuan Yang, Yaodong Yu, Jarosław Błasiok, and Preetum Nakkiran. What you see is what you get: Principled deep learning via distributional generalization. *Advances in Neural Information Processing Systems*, 35, 2022.

[108] Bogdan Kulynych, Hsiang Hsu, Carmela Troncoso, and Flavio P Calmon. Arbitrary decisions are a hidden cost of differentially-private training. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT)*, 2023.

[109] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[110] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

[111] Qi Lei, Lingfei Wu, Pin-Yu Chen, Alex Dimakis, Inderjit S Dhillon, and Michael J Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Proceedings of Machine Learning and Systems*, 2019.

[112] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *29th USENIX Security Symposium*, 2020.

[113] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *CODASPY*, 2021.

[114] Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 2012.

[115] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *IJCAI*, 2018.

[116] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

[117] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015.

[118] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[119] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020.

[120] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*, 2019.

[121] Daniel Lowd and Christopher Meek. Adversarial learning. In *KDD*, 2005.

[122] Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

[123] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR*, 2018.

[124] Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International Conference on Machine Learning*. PMLR, 2020.

[125] Yael Mathov, Eden Levy, Ziv Katzir, Asaf Shabtai, and Yuval Elovici. Not all datasets are born equal: On heterogeneous tabular data and adversarial examples. *Knowledge-Based Systems*, 2022.

[126] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *FAT**, 2019.

[127] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019.

[128] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[129] Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization. *arXiv preprint arXiv:2009.08092*, 2020.

[130] Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[131] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, 2018.

[132] Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[133] Boel Nelson and Jenni Reuben. SoK: Chasing accuracy and privacy, and catching both in differentially private histogram publication. *Trans. Data Priv.*, 2020.

[134] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[135] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

[136] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with Renyi differential privacy. In *International Conference on Learning Representations*, 2022.

[137] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[138] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Euro S&P*, 2016.

[139] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[140] 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*, 2011.

[141] Guillaume Perez, Sebastian Ament, Carla Gomes, and Michel Barlaud. Efficient projection algorithms onto the weighted l1 ball. *Artificial Intelligence*, 306, 2022.

[142] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ML attacks in the problem space. *IEEE S&P*, 2020.

[143] Ira Pohl. Heuristic search viewed as path finding in a graph. *Artif. Intell.*, 1970.

[144] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. 2014.

[145] David Pujol, Ryan McKenna, Satya Kuppam, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair decision making using privacy-protected data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2020.

[146] Sofya Raskhodnikova, Adam Smith, Homin K Lee, Kobbi Nissim, and Shiva Prasad Kasiviswanathan. What can we learn privately. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, 2008.

[147] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2020.

[148] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1), 2012.

[149] Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *IEEE Transactions on Information Theory*, 2019.

[150] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*. PMLR, 2019.

[151] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[152] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2019.

[153] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security (NDSS) Symposium*, 2019.

[154] Amartya Sanyal, Yaxi Hu, and Fanny Yang. How unfair is private learning? In *Uncertainty in Artificial Intelligence*. PMLR, 2022.

[155] James C Scott. *Seeing like a state: How certain schemes to improve the human condition have failed*. Yale University Press, 2020.

[156] Howard J Seltman. Experimental design and analysis. *Department of Statistics at Carnegie Mellon (Online Only)*, 2009.

[157] Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022.

[158] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

[159] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 2010.

[160] Mahmood Sharif, Lujo Bauer, and Michael K Reiter. On the suitability of l_p-norms for creating and preventing adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[161] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE symposium on security and privacy (SP)*, 2017.

[162] Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.

[163] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2021.

[164] Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory*, 2020.

[165] Roni Stern, Rami Puzis, and Ariel Felner. Potential search: A bounded-cost search algorithm. In *ICAPS*, 2011.

[166] Roni Stern, Ariel Felner, Jur van den Berg, Rami Puzis, Rajat Shah, and Ken Goldberg. Potential-based bounded-cost search and anytime non-parametric A*. *Artificial Intelligence*, 2014.

[167] Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. In Marina Meila and Tong Zhang, editors, *ICML*, 2021.

[168] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[169] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

[170] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 10–19, 2019.

[171] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

[172] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2018.

[173] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

[174] Daniël Vos and Sicco Verwer. Efficient training of robust decision trees against adversarial examples. In Marina Meila and Tong Zhang, editors, *ICML*, 2021.

[175] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31, 2017.

[176] Hao Wang, Rui Gao, and Flavio P Calmon. Generalization bounds for noisy iterative algorithms using properties of additive noise channels. *Journal of Machine Learning Research*, 24, 2023.

[177] Yu-Xiang Wang. Per-instance differential privacy. *arXiv preprint arXiv:1707.07708*, 2017.

[178] Yu-Xiang Wang, Jing Lei, and Stephen E Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *The Journal of Machine Learning Research*, 2016.

[179] Yutong Wang, Yufei Han, Hongyan Bao, Yun Shen, Fenglong Ma, Jin Li, and Xiangliang Zhang. Attackability characterization of adversarial evasion attack on discrete data. In *KDD*, 2020.

[180] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

[181] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 2010.

[182] Jamelle Watson-Daniels, David C Parkes, and Berk Ustun. Predictive multiplicity in probabilistic classification. In *AAAI*, 2023.

[183] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, 2010. doi: 10.25080/Majora-92bf1922-00a.

[184] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[185] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[186] Laurence A. Wolsey. Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Math. Oper. Res.*, 1982.

[187] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

[188] Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *ICML*, 2019.

[189] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21, 2018.

[190] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017.

[191] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017.

[192] Depeng Xu, Wei Du, and Xintao Wu. Removing disparate impact on model accuracy in differentially private stochastic gradient descent. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

[193] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. Greedy attack and Gumbel attack: Generating adversarial examples for discrete data. *JMLR*, 2020.

[194] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018.

[195] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

[196] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In *Proceedings of the 39th International Conference on Machine Learning, ICML*, 2022.

[197] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019.

[198] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[199] Jingzhao Zhang, Aditya Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. Coping with label shift via distributionally robust optimisation. *arXiv preprint arXiv:2010.12230*, 2020.

[200] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[201] Xiao Zhang and David Evans. Cost-sensitive robustness against adversarial examples. In *International Conference on Learning Representations*, 2019.

[202] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017.

[203] Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *The Journal of Machine Learning Research*, 23(1), 2022.

Bogdan Kulynych

Website: https://kulyny.ch · Email: bogdan@kulyny.ch

Education

2018 — 2023	**EPFL,** Switzerland *Ph.D. in Computer Science*
2016 — 2017	**Polytechnic University of Madrid,** Spain. *M.Sc. in Software and Systems.* GPA: 9.3 / 10
2012 — 2013	**Jagiellonian University,** Poland. *Erasmus Mundus Exchange Student.* GPA: 4.8 / 5
2010 — 2015	**National University of Kyiv Mohyla Academy,** Ukraine. *Bachelor's in Applied Mathematics.* GPA: 89 / 100

Work Experience

2022.07 — 2022.12	**Harvard University,** Boston, USA. *Visiting Fellow*
2018.02 — 2023.09	**EPFL SPRING Lab,** Lausanne, Switzerland. *Ph.D. Researcher*
2017.10 — 2018.01	**Google,** Mountain View, CA. *Software Engineering Intern*
2016.10 — 2017.10	**IMDEA Software Institute,** Madrid, Spain. *Research Assistant*
2016.06 — 2016.09	**Google,** Los Angeles, CA. *Software Engineering Intern*
2015.10 — 2016.01	**Google,** Mountain View, CA. *Software Engineering Intern*
2014.07 — 2014.08	**CERN,** Geneva, Switzerland. *Openlab Summer Student*

Scholarships and Awards

2022	**Scholar Award,** Neural Information Processing Systems (NeurIPS) conference
2021	**1st place at the Algorithmic Bias Challenge,** Twitter
2017	**Travel grant,** 7th Bar-Ilan University Winter School on Cryptography
2017	**Travel grant,** Privacy Enhancing Technologies Symposium (PETS)
2014	**Best Technology Award,** CERN WebFest Hackathon
2012	**EMERGE Erasmus Mundus Scholarship,** The EU

Academic Service and Co-organized Events (Selected)

2023	**NeurIPS** (regular and ethics reviewer), **ICML, FAccT, The Web Conference** (reviewer)
2022	**NeurIPS** (regular and ethics reviewer), **FAccT** (reviewer)
2021	**NeurIPS** (ethics reviewer)
2020	**Justice and Technology Table at LSE** (advisory board member) **Participatory Approaches to Machine Learning Workshop at ICML** (co-organizer)

Teaching

2018 – 2022	**EPFL,** Switzerland. *Teaching assistant* *COM-301 Security and Privacy (Bachelor's level).* *COM-402 Information Security and Privacy (Master's level).* *CS-523 Advanced Topics in Privacy-Enhancing Technologies (Master's level).*
2019	**Lviv Data Science Summer School,** Ukraine. *Lecturer.* *Security and Privacy of Machine Learning (6-hour course).*

Publications (Selected)

* denotes equal contribution.

2023 **Arbitrary Decisions are a Hidden Cost of Differentially Private Training**
B. Kulynych, H. Hsu, C. Troncoso, F. du Pin Calmon
To appear in the proceedings of ACM Fairness, Accountability, and Transparency (FAccT)

Adversarial Robustness for Tabular Data through Cost and Utility Awareness
K. Kireev*, B. Kulynych*, C. Troncoso
To appear in the proceedings of Network and Distributed Systems Security (NDSS)
Presented at the NeurIPS 2022 Safe ML workshop.

2022 **What You See is What You Get: Principled Deep Learning via Distributional Generalization**
B. Kulynych*, Yao-Yuan Yang*, Y. Yu, J. Blasiok, P. Nakkiran
Published in the proceedings of Neural Information Processing Systems (NeurIPS)
Presented at the NeurIPS 2022 Safe ML workshop.

Disparate Vulnerability to Membership Inference Attacks
B. Kulynych, M. Yaghini, G. Cherubin, M. Veale, C. Troncoso
Published in the proceedings of Privacy Enhancing Technologies Symposium (PETS)
Presented at Privacy Preserving Machine Learning Workshop at ACM CCS 2019 conference

2021 **Exploring Data Pipelines through the Process Lens: a Reference Model for Computer Vision**
A. Balayn, B. Kulynych, S. Gürses
Presented at “Beyond Fair Computer Vision" Workshop at CVPR 2021

Adversarial for Good? How the Adversarial ML Community’s Values Impede Socially Beneficial Uses of Attacks
K. Albert*, M. Delano*, B. Kulynych*, R. Shankar Siva Kumar*
Presented at “A Blessing in Disguise: The Prospects and Perils of Adversarial Machine Learning" Workshop at ICML 2021

2020 **Protective Optimization Technologies**
B. Overdorf*, B. Kulynych*, E. Balsa, C. Troncoso, S. Gürses
Published in the proceedings of the ACM Fairness, Accountability, and Transparency (FAccT)

Localisation par le réseau mobile
G. Cherubin, B. Kulynych, M. Le Tilly, C. Troncoso
Published in the bulletin.ch Magazine

2019 **zksk: A Library for Composable Zero-Knowledge Proofs**
W. Lueks, B. Kulynych, J. Fasquelle, S. Le Bail-Collet, C. Troncoso.
Published in the proceedings of the Workshop on Privacy in the Electronic Society (WPES)

2018 **Questioning the Assumptions Behind Fairness Solutions**
B. Overdorf*, B. Kulynych*, E. Balsa, C. Troncoso, S. Gürses
Presented at “Critiquing and Correcting Trends in ML” Workshop at NeurIPS

Evading Classifiers in Discrete Domains with Provable Optimality Guarantees
B. Kulynych, J. Hayes, N. Samarin, C. Troncoso
Presented at the Workshop on Security and Privacy in ML at NeurIPS

ClaimChain: Improving the Security and Privacy of In-band Key Distribution for Messaging
B. Kulynych*, M. Isaakidis*, Wouter Lueks, George Danezis, Carmela Troncoso
Published in proceedings of the Workshop on Privacy in the Electronic Society (WPES)