BENCHMARKING AND ENHANCING THE UTILITY OF DIFFERENTIAL PRIVACY FOR DATA MINING APPLICATIONS

JIAWEI DUAN

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Electrical and Electronic Engineering

Benchmarking and Enhancing the Utility of Differential Privacy for Data Mining Applications

Jiawei Duan

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
September 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Jiawei Duan

Abstract

Differential Privacy (DP) offers robust guarantees for protecting individual data against malicious attacks in both industrial sectors (e.g., Apple and Google) and administrative sectors (e.g., the U.S. Census Bureau). In general, DP allows for efficient statistical analysis while safeguarding privacy, making it widely adopted in various data mining tasks such as frequency/mean estimation, private data publication, and private learning. However, there exists a trade-off between utility and privacy: enhancing one typically compromises the other. Despite significant efforts to mitigate this trade-off, critical limitations persist. Some solutions achieve high utility but are tailored to specific DP mechanisms or data mining tasks, thus lacking generality. Conversely, more general solutions often fail to deliver superior utility. This creates a dilemma where achieving both generality and effectiveness simultaneously remains challenging.

The works in this thesis together compose a platform that theoretically benchmarks and generally enhances the utilities of various DP mechanisms in two prevalent data mining scenarios: statistical analysis and model training. The main contributions of this thesis are divided into three chapters, organized in a top-down order: high-dimensional statistics estimation [7,51,84,88,104], centralized learning [6,82,85,160], and federated learning [150].

In the first chapter, we present LDPTube, an analytical toolbox that generalizes and enhances DP mechanisms for high-dimensional mean estimation. Specifically,

we leverage the Central Limit Theorem (CLT) [43, 115], one of the most recognized theorems in statistics, to describe the mean square errors (MSEs) of various DP mechanisms. To optimize their MSEs, HDR4ME* uses regularizations to eliminate excessively noisy data, thereby achieving better utilities in high-dimensional mean estimation. The second chapter focuses on the utilities of private centralized learning. Here, we introduce GeoDP, a framework that first theoretically derives the impact of DP noise on model efficiency. Our analysis reveals that the existing perturbation methods introduce biased noise to the gradient direction, resulting in a sub-optimal training process. GeoDP addresses this issue by adding unbiased noise to the gradient direction, thereby improving model utilities. In the final chapter, we propose LDPVec, which theoretically analyzes and enhances model utility in federated learning under various DP mechanisms. Similar to mean estimation, the global aggregation step in federated learning averages noisy gradients from each local party, allowing the CLT to effectively describe model utilities. We observe that preserving the gradient direction is crucial, while the perturbed gradient magnitude can be adjusted through fine-tuning the learning rate or clipping. Consequently, LDPVec optimizes model efficiency by allocating (d-1)/d of the privacy budget to the gradient direction and 1/d to the gradient magnitude.

Publications Arising from the Thesis

1. J. Duan, Q. Ye and H. Hu, "Utility Analysis and Enhancement of LDP mechanisms in High- dimensional Space," *in IEEE International Conference on Data Engineering (ICDE), 2022.*

2. J. Duan, Q. Ye, H. Hu and X. Sun, "LDPTube: Theoretical Utility Benchmark and Enhancement for LDP Mechanisms in High-dimensional Space," *in IEEE Transactions on Knowledge and Data Engineering (TKDE), 2024.*

3. X. Sun, Q. Ye, H. Hu, J. Duan, Q. Xue, T. Wo and J. Xu, "PUTS: Privacy-Preserving and Utility-Enhancing Framework for Trajectory Synthesization," *in IEEE International Conference on Data Engineering (ICDE), 2024.*

4. J. Duan, Q. Ye, H. Hu and X. Sun, "Analyzing and Enhancing LDP Perturbation Mechanisms in Federated Learning," under Review in *IEEE Transactions on Knowledge and Data Engineering (TKDE), 2025.*

5. J. Duan, Q. Ye, H. Hu, and X. Sun. "Analyzing and Optimizing Perturbation of DP-SGD Geometrically," under Review in *International Conference on Learning Representations (ICLR), 2025.*

6. X. Sun, Q. Ye, H. Hu, J. Duan, Q. Xue, T. Wo and J. Xu, "Generating Location Traces with Semantic- Constrained Local Differential Privacy," under Review

in *IEEE Transactions on Information Forensics and Security (TIFS), 2025.*

Acknowledgments

Five years ago, I came to Hong Kong to pursue a Master degree. Regardless of my lack of research experiences, Prof. Hu Haibo generously provided me with a research-assistant position, which allowed me to take a first glance at research. Back then, I greedily consumed knowledge and looked forward to a doctoral study, while neglecting the cost of a PhD degree.

In my first year of doctoral study, I was assigned to a company in Hong Kong Science Park and took part in some industrial programs. I hereby regard my team leader, Dr. Tom Chan, for the guidance on my profession and the opportunity for me to grow up. As a very dedicated man, he inspired me to never give up and do whatever I can for success. I always value experiences when we worked together and take them as my life treasure.

In the rest of doctoral study, I returned to Prof. Hu's lab and was primarily supervised by him. I would like to show my utmost regard to him. Prof. Hu is one of the most distinguished researcher that I have ever seen. As the mentor who opened the gate of research for me, his wisdom is much beyond my understanding. Once in a while, I doubted his judgments and questioned his concerns. It always turned out that I was just too slow to understand his intelligence. Inspired by his rigorous attitude and unconditional supports, I always had the courage to challenge myself and mine my own potential. Apart from the academic ability, he also instructed me to cast aside the arrogance and never stop the pursuit of excellence. He is the one who changed

me, and I can never be more grateful for what he has devoted.

In addition, I extend my acknowledgment to Dr. Ye Qingqing, Dr. Sun Xinyue, Dr. Zheng Huadi, Dr. Fu Yue, Ms. Du Rong, and Mr. Huang Chao. Dr. Ye Qingqing is the one who led me to Differential Privacy, and I always admire her rigorousness, intelligence and warm heart. As a pioneer in the scope, she always reminds me to get better and better in my academic career. Dr. Sun Xinyue, as my best co-author, worked with me to overcome countless academic challenges. Dr. Zheng Huadi, as a senior PhD in our group, keeps encouraging me to be optimistic about the future and never underestimate myself. Dr. Fu Yue and Ms. Du Rong supported me to spend the most naive period of the doctoral study, and I always honor the friendship with them. Mr. Huang Chao, as a very helpful and supportive guy, is my best friend in Hong Kong. I will always remember happy moments spent with him.

Last but not least, I express my utmost gratitude to my family and my love of life, Miss Huang Zirong. My parents and sister have no hesitation to support my study in Hong Kong, both financially and emotionally. Without their supports, there is no PhD degree. Besides, I am so lucky to have Miss Huang, who is blessed with all virtues of human life. The union of her love brings me ecstasy, which is so great that I would like to sacrifice all the rest of my life just for a tiny moment of this joy.

Table of Contents

Abstract **i**

Publications Arising from the Thesis **iii**

Acknowledgments **v**

List of Figures **xii**

List of Tables **xiv**

1 Introduction **1**

1.1 Private Statistics Estimation 2

1.2 Private Machine/Deep Learning 3

1.3 Crossover Applications 4

1.4 Main Objectives and Organization of Thesis 5

2 Literature Review **8**

2.1 Locally Differentially Private (LDP) Statistics Estimation 8

2.1.1 Mean Estimation by LDP 9

2.1.2 High-Dimensional LDP 10
2.2 Private Learning 10
2.2.1 Differential Privacy (DP) 11
2.2.2 Stochastic Gradient Descent (SGD) 11
2.2.3 Differentially Private Stochastic Gradient Descent (DP-SGD) . 12
2.2.4 Inference Attacks (IA) 13
2.3 Crossovers of Private Learning and Statistics Estimation 14
2.3.1 Federated Learning 14
2.3.2 Integrating LDP with FL (Federated LDP-SGD) 15

3 Preliminary **16**
3.1 Basic Concepts 16
3.1.1 Differential Privacy 16
3.1.2 Local Differential Privacy 18
3.1.3 Differentially Private Stochastic Gradient Descent (DP-SGD) . 20
3.1.4 Federated Averaging and Optimization Techniques 21
3.2 Problem Definition 22
3.2.1 High-dimensional Statistics Estimation 22
3.2.2 Private Learning Analysis and Enhancement 24
3.2.3 Model Efficiency of Federated LDP-SGD 26

4 Analyzing and Enhancing LDP Mechanisms in High-dimensional Space **28**

4.1 A General Analytical Framework 32
4.1.0.1 Saving The Burden of Choosing Parameters 37
4.1.0.2 Computational Saving Under Population Variation in Practice 39
4.1.1 Approximation Error of Theorem 1 42
4.1.2 A Case Study: How to Benchmark Piecewise Mechanism and Square Wave Mechanism in High-Dimensional Space? 44
4.1.3 Utility Analysis in Personalized Local Differential Privacy (PLDP) 49
4.2 HDR4ME*: High-dimensional Re-calibration for Mean Estimation . . 52
4.2.1 Regularization: Diminishing Utility Deterioration in High-dimensional Space 53
4.2.2 HDR4ME*—High Dimensional Re-calibration for Mean Estimation 53
4.2.3 High-dimensional Re-calibration for Frequency Estimation . . 62
4.3 Empirical Results 62
4.4 Summary 70

5 Analyzing and Optimizing Perturbation of DP-SGD Geometrically 72

5.1 Deficiency of DP-SGD: a gap between directional SGD and numerical DP 77
5.2 Geometric perturbation: GeoDP 81
5.2.1 Hyper-spherical Coordinate System 82
5.2.2 GeoDP—Geometric DP Perturbation for DP-SGD 83

5.2.3 Comparison between GeoDP and Traditional DP: Efficiency and Privacy . . . 88
5.2.3.1 Efficiency Comparison . . . 88
5.2.3.2 Privacy Comparison . . . 93
5.3 Experimental results . . . 95
5.3.1 Experimental Setup . . . 95
5.3.1.1 Datasets and Models . . . 95
5.3.1.2 Competitive Methods . . . 97
5.3.2 GeoDP vs. DP: Accuracy of Descent Trend . . . 97
5.3.3 GeoDP vs. DP: Logistic Regression . . . 99
5.3.4 GeoDP vs. DP: Deep Learning . . . 102
5.3.5 GeoDP versus DP: The Defense on MIA . . . 104
5.4 Summary . . . 105

6 Analyzing and Enhancing LDP Perturbation in Federated Learning 106

6.1 A General Analytical Framework for Federated LDP-SGD . . . 108
6.1.1 Overview of Federated LDP-SGD . . . 109
6.1.2 Model of Federated LDP-SGD . . . 110
6.1.3 A General Analytical Framework for Federated LDP-SGD . . . 114
6.1.4 A Case Study: Benchmarking LDP Mechanisms in Federated Logistic Regression . . . 117
6.2 LDPVec: Enhancing Federated LDP-SGD from A Geometric Perspective 120
6.2.1 LDPVec—Vectorized Perturbation for Federated LDP-SGD . . . 121

6.3 Experimental Evaluation 125
6.3.1 Datasets and Models 125
6.3.2 Parameter Settings for Federated LDP-SGD 127
6.3.3 Effectiveness of LDPVec in Machine Learning 127
6.3.4 Effectiveness of LDPVec in Deep Learning 128
6.4 Summary 131

7 Conclusion and Future Works **132**
7.1 Conclusion 132
7.1.1 Mean Estimation in High-Dimensional Spaces by LDP Mechanisms 132
7.1.2 Optimization of DP-SGD 133
7.1.3 Efficiencies of LDP Mechanisms in Federated Learning 133
7.2 Future Work 133

References **135**

List of Figures

4.1 Overview of LDPTube. 31

4.2 Regularization in two dimensions. 54

4.3 Analysis vs. experimental results on Uniform dataset under regular LDP (d=5,000). 64

4.4 Analysis vs. experimental results on Uniform dataset under personalized LDP (d=2,000). 65

4.5 Analysis vs. experimental results in our case study. 66

4.6 MSE on various datasets and dimensions 68

4.7 MSE on COV-19 dataset with various dimensions. 69

4.8 MSE on Real Datasets. 70

5.1 Comparing MSEs of GeoDP and DP on preserving directions and values of gradients under synthetic dataset (composed of gradients from CNN training, as introduced in Section 5.3.1). While θ and g label the MSE of perturbed directions and gradients themselves, experimental results confirm that GeoDP achieves smaller MSEs on perturbed directions (i.e., the red line is below the black one), while sacrificing the accuracy of perturbed gradients (i.e., the green lien is above the blue one). In general, GeoDP better preserves directions of gradients while traditional DP only excels in preserving numerical values of gradients. 74

5.2 Coordinates Conversions in Three-dimensional Space. 84

5.3 GeoDP vs. DP on Preserving Gradients under Various Parameters on Synthetic Dataset 100

5.4 The Effectiveness of Bounding Factor 101

5.5 GeoDP versus DP on Logistic Regression under MNIST dataset . . . 101

6.1 Overview of Federated LDP-SGD. 110

6.2 Global Loss of Logistic Regression on MNIST under Various ϵ, E and N 129

6.3 Global Loss of Multilayer Perceptron on CIFAR-10 130

List of Tables

4.1 Notations 30

4.2 Probabilities for the supremum to hold in one dimension 46

5.1 Frequently-used notations 76

5.2 GeoDP vs. DP on CNN under MNIST Dataset: Test Accuracy 102

5.3 GeoDP vs. DP on ResNet under CIFAR-10 Dataset: Test Accuracy . 103

5.4 GeoDP versus DP on ML-Doctor: Attack Accuracy 104

6.1 Parameters for Logistic Regression on MNIST 126

6.2 MLP Architecture and Parameters on CIFAR-10. 126

6.3 CNN Architecture and Parameters on CIFAR-10. 126

6.4 Perturbation Comparison under Federated CNN: Testing Accuracy ($N = 1000$, $E = 200$, $\beta = 0.05$) 130

Chapter 1

Introduction

Data mining techniques, which discovers meaningful patterns, correlations, anomalies, and trends in large datasets, can assist organizations to make data-driven decisions to improve business processes, enhance customer experiences, and gain a competitive edge. Composed of data correlation analysis, dimensionality reduction, machine/deep learning models, and optimization methods, data mining has been flourishing in various sectors, such as finance, healthcare, energy, and scientific researches. Nonetheless, privacy can be compromised when data is accessed in data mining, and data leakage may lead to serious reputational damage and financial loss. For example, it was revealed in 2018 that Cambridge Analytica, a political consulting firm, had utilized private data from millions of Facebook users without their consent to provide political advantages for clients. Also, Facebook itself faced multiple lawsuits and regulatory scrutiny, leading to a five-billion-dollar fine for privacy violations. Both examples, which indicate how privacy leakage affects an organization's long-term viability and market position, stress the significance of privacy preservation in data mining techniques. Among various privacy solutions, differential privacy (DP) has at least five unique advantages. First, subject to strong mathematical assurances, it provides robust guarantees, regardless of how and where data is utilized. Second, it demonstrates

strong scalability to complex systems. The way that DP linearly adds noise ensures its effectiveness across various algorithms and datasets, making it versatile for different applications. Third, DP can control the loss of privacy due to the mathematical definition. Fourth, random noise introduced by DP enables strong defense against various attacks. Last but not least, the integration of DP into existing algorithms does not significantly increase complexity. Overall, DP is one of the most powerful tool of privacy preservation in data mining.

However, there exists a trade-off between data utility and privacy [140]. In general, adding more noise, which certainly improves privacy level, otherwise obfuscates the original algorithm and therefore sabotages utility. As such, how to balance these two is perhaps the most vital problem in DP-related data mining. To fully address this problem, this thesis presents a comprehensive system, which can analyze and enhance utilities of DP in various data mining tasks while maintaining the same privacy level. In terms of three common scenarios (i.e., statistics estimation, machine/deep learning, and their crossover applications) of data mining, this system proposes respective set of solutions.

1.1 Private Statistics Estimation

In data mining, statistics estimations, including mean and frequency estimation, are fundamental tasks and therefore worth studying. By these two basic estimations, we are allowed to analyze patterns of datasets and accordingly implement more complex algorithms. In this thesis, we systematically analyze and enhance the utility of differentially private statistics estimation. In specific, via Central Limit Theorem (CLT) [43, 115], we are able to derive distributions of Mean Square Error (MSE) of any DP mechanism in various datasets. That is, we obtain results of experiments without conducting any experiment [30,31]. Even under extreme cases (e.g., the population is $1,000$ and one mediocre DP mechanism Laplace [36]), the error of analysis

is no more than 1.5%. This analysis further instructs how to enhance utility. By regularization, we successfully reduce dimensionality of high-dimensional data and thus random noise per dimension. Extensive analysis and experiments the effectiveness of this set of solutions. Our major contributions on private statistics estimation are summarized as follows:

- We bring forward a non-parametric analytical framework to measure the utilities of LDP mechanisms in high-dimensional space. This framework not only provides a theoretical baseline to benchmark existing and future LDP mechanisms, but also serves as a platform to compare their theoretical utilities in high-dimensional space.
- We propose a re-calibration protocol *HDR4ME** to enhance high-dimensional mean estimation and prove its superiority to the baseline. In particular, this protocol can be further extended to frequency estimation.
- Based on both synthetic and real datasets, we conduct extensive experiments to validate our framework and evaluate our protocol for three state-of-the-art high-dimensional LDP mechanisms. Results show that the theoretical benchmark is consistent with the experimental results, and our protocol generally enhances the utilities.

1.2 Private Machine/Deep Learning

While statics directly mines patterns from datasets, some insights are subtle and require machine/deep learning to derive. In recent years, private learning [1], which adds noise to gradients while performing training, has been prevalent in data mining by enabling the discovery of patterns and insights from large datasets. However, DP noise still obfuscates the training process, leading even more serious model utility degradation. To address this problem, our system proposes a general solution GeoDP,

to analyze and enhance utilities of various DP mechanisms in different learning models. Extensive analysis and experiments confirm that the utility benefit brought by GeoDP is rather huge under the same privacy level. Major contributions on private learning are as follows:

- To the best of our knowledge, we are the first to prove that the perturbation of traditional DP-SGD is actually sub-optimal from a geometric perspective.
- Within the classic DP framework, we propose a geometric perturbation strategy *GeoDP* to directly add the noise on the direction of a gradient, which rigorously guarantees a better trade-off between privacy and efficiency.
- Extensive experiments on public datasets as well as prevalent AI models validate the generality and effectiveness of GeoDP.

1.3 Crossover Applications

There also exists a scenario where both statistics estimation and model training are effective. Locally differentially private federated learning (LDP-FL), which adds noises to local models for privacy while performing global aggregation to make noises canceled out for better utility, can be considered as an intersection of private learning and statistics estimation. This system proposes LDPVec to analyze and enhance the model utility of LDP-FL, respectively. In specific, LDPVec analyzes the utility of global aggregation from a perspective of mean estimation while enhancing model utility by utilizing geometric property of a gradient. Our major contributions on this intersection are as follows:

- To the best of our knowledge, this is the first general analytical framework to measure LDP mechanisms in federated learning. This framework can not

only serve as a benchmark to compare various LDP mechanisms in federated LDP-SGD, but also point out the direction of future optimization.

- We propose a geometric perturbation strategy *LDPVec*, which optimizes performances of various LDP mechanisms in federated SGD.

- Extensive experiments on real datasets, popular machine learning models, and three state-of-the-art LDP mechanisms are conducted to validate the generality and effectiveness of both framework and strategy. All results unanimously show that the theoretical analysis is consistent with the experimental results, and our geometric perturbation strategy significantly improves model efficiencies in practice.

1.4 Main Objectives and Organization of Thesis

Overall, this thesis proposes a systematical solution to analyzing and enhancing performances of various DP mechanisms in general data mining scenarios. In specific, this thesis is organized as presented in the following chapters:

Chapter 2: This chapter comprehensively reviews DP mechanisms in data mining techniques, including private mean/frequency estimation, differentially private stochastic gradient descent (DP-SGD), and federated locally differentially private stochastic gradient descent (federated LDP-SGD).

Chapter 3: This chapter provides backgrounds and preliminaries throughout this thesis.

Chapter 4: In this chapter, we first propose an analytical framework that generalizes LDP mechanisms and derives their utilities in high-dimensional space, namely the probability density function of the deviation between the estimated mean and the true mean. The framework can serve as a benchmark to compare

the utilities of various LDP mechanisms without conducting any experiment. Furthermore, our analysis shows the sub-optimality of the naive aggregation method of all LDP mechanisms — the utility deterioration is attributed to the overwhelming noise caused by diluted privacy budget in high-dimensional space. As such, our second contribution in this paper is a one-off, non-iterative re-calibration protocol *HDR4ME* (acronym for High-Dimensional Re-calibration for Mean Estimation). Through regularization and proximal gradient descent, this protocol re-calibrates the aggregated mean obtained from any LDP mechanism by suppressing the overwhelming noise and thus enhances its utility. Without any change on the LDP mechanism itself, *HDR4ME* can be used as a general optimizer of existing LDP mechanisms in high-dimensional space.

On this basis, we propose a toolbox *LDPTube* (acronym for Theoretical Utility Benchmark and Enhancement for LDP mechanisms) for practically utility benchmark and maximization. First, it consists of a non-parametric benchmark in high-dimensional space, which relieves the burden of the data collector for choosing the supremum from an unknown distribution. It not only provides mean square error as the analytical result, but also shows the population breakpoint where one LDP mechanism outperforms the other in terms of MSE. In addition, for generality, this benchmark can also apply to personalized LDP, where users are free to choose privacy budgets and privacy regions (i.e., the domain where one's original data remains anonymous). Besides the benchmark, the existing *HDR4ME* re-calibration protocol in [30] does not perform well under low error, which is rather common in practice. In LDPTube, we next present a utility maximization protocol, namely *HDR4ME**, to adaptively and optimally select between L_1-, L_2-regularization, and no-regularization for a LDP mechanism to maximize utilities in high-dimensional space.

Chapter 5: In this chapter, we propose a geometric perturbation strategy GeoDP to address the inefficiency of DP mechanism in various learning tasks.

First, we theoretically derive the impact of DP noise on the efficiency of DP-SGD. Proved by this fine-grained analysis, the perturbation of DP-SGD, which introduces biased noise to the direction of a gradient, is actually sub-optimal. Inspired by this, we propose a geometric perturbation strategy *GeoDP* which perturbs both the direction and the magnitude of a gradient, so as to relieve the noisy gradient direction and optimize model efficiency with the same DP guarantee.

Chapter 6: In this chapter, we analyze and improve model utilities of federated LDP-SGD by first proposing an analytical framework that generalizes federated LDP-SGD and derives the impact of LDP noise on the federated training process, in terms of the model efficiency. Then we show that this framework can serve as a benchmark to compare model efficiencies of federated LDP-SGD under various LDP mechanisms. An interesting observation is that while existing works preserve the gradient itself, our analysis points out that only its direction is necessary for gradient descent. As such, existing LDP-SGD strategy is sub-optimal, as it wastes privacy budget to preserve the magnitude of gradient. Motivated by this, our second contribution is a geometric perturbation strategy *LDPVec* to optimize the training process. While focusing on preserving directional information, *LDPVec* only perturbs the direction of a gradient, and rearranges LDP noise to better preserve directional information. This strategy can generally enhance federated LDP-SGD under various LDP mechanisms.

Chapter 7: We conclude the outcomes of this thesis and propose some new directions for future works in private data mining.

Chapter 2

Literature Review

In this chapter, we comprehensively review differential privacy in various data mining techniques. In specific, we review private statistics estimation, private learning, and their crossovers.

2.1 Locally Differentially Private (LDP) Statistics Estimation

Dwork *et al.* [35] formally present the definition of *differential privacy* (DP) and propose the first DP mechanism, i.e., *Laplace mechanism*. As for the local setting, Evfimievski *et al.* [39] are among the first to introduce differential privacy at the side of individuals. Then Raskhodnikova *et al.* [105] design a locally private mechanism γ-*amplification randomizer*. Later on, Duchi *et al.* [32] study the trade-off between local privacy budget and estimation utility, and derive bounds for *local differential privacy* (LDP). LDP has been widely adopted in different domains, including itemset mining [138], marginal release [26, 166], time series data release [154], graph data analysis [121,152,153], key-value data collection [56,155,156] and private learning [168,

169]. The most relevant problems to this paper include two aspects, namely, mean estimation by LDP and high-dimensional LDP.

2.1.1 Mean Estimation by LDP

Dwork *et al.* [35] initially propose *Laplace mechanism* for mean estimation for centralized DP, which can also be applied to the local setting. Afterwards, several LDP frameworks, such as a variant of *Laplace mechanism* referred to as *SCDF* [120] and *Staircase mechanism* [49], perturb values with less noise. Note that the perturbed values of these mechanisms range from negative to positive infinity, so they are classified as *unbounded mechanisms* in this paper. On the contrary, *bounded mechanisms* perturb values into a finite domain. Duchi *et al.* [33] present one whose outputs are binary. To overcome the shortcoming of binary output, Wang *et al.* [134] propose *Piecewise mechanism* and *Hybird mechanism*. With continuous and bounded outputs, their utilities are improved. More recently, Li *et al.* [83] propose *square wave mechanism* where the perturbation is more centered than Piecewise, and the utility is therefore superior. As regards frequency estimation, there are a family of Randomized Response (RR). Considering that primitive RR is only capable of binary values, Kairouz *et al.* [69] present *k-RR* for multiple variables. With introduction of cohort-based hashing, Kairouz *et al.* [67] also bring up *O-RR* and *O-PAPPOR*. Bassily *et al.* [10] then present an efficient protocol *SHist* to diminish the communication cost. On this basis, Bassily *et al.* [9] latter design *TreeHist* and *Bitstogram* as improved LDP heavy hitters. To generalize RR protocols and choose parameters for optimization, Wang *et al.* [137] introduce a general framework, which includes prevalent RR protocols and provides shareable error metrics.

In industry, Apple [25, 27, 124] applies *Count Mean Sketch* techniques to macOS and iOS for collecting users' browser hints. Meanwhile, Google [38, 40] initiates *RAPPOR* to collect and analyze users' strings. Microsoft [28] also proposes such frameworks

1-bit mechanisms and α-*point* rounding scheme for *count number* estimation.

2.1.2 High-Dimensional LDP

The most critical challenge to adopt LDP in high-dimensional space is the utility degradation, a.k.a., the dimensionality curse. In general, there are two streams of methodology to cope with it. One is dimensionality reduction. As for non-local privacy data publication, Ren *et al.* [106] study frequency estimation based on *Lasso Regression* and *EM* algorithm. By *principal components analysis* (PCA), Ge *et al.* [48] propose *DPS-PCA* for interactive LDP while Wang *et al.* [133] consider PCA for non-interactive LDP. Besides, Bassily [8] studies linear queries estimation in high-dimensional LDP. The other methodology is correlation-based privacy budget allocation. Chatzikokolakis *et al.* [21] use metric d_h to measure the similarity between two dimensions in DP. Larger d_h indicates lower similarity, which requires more privacy budget in those dimensions. Alvim *et al.* [2] extend this metric to LDP. Similarly, Li *et al.* [81] calculate the respective information entropy of all dimensions while Du *et al.* [29] use covariance of different dimensions to allocate privacy budget accordingly.

It is noteworthy that almost all these works have limited the application scope in specific scenarios. Furthermore, many solutions have high computational cost at the user side [2,21,33,48,133]. This thesis, on the other hand, enhances high-dimensional LDP mean estimation by only involving the data collector. In addition, it is a general optimization that is irrespective of the LDP mechanisms.

2.2 Private Learning

In this section, we review related works from three aspects: DP, SGD and their crossover works DP-SGD.

2.2.1 Differential Privacy (DP)

DP [36,140] is a framework designed to provide strong privacy guarantees for datasets whose data is used in data analysis or machine learning models. It aims to allow any third party, e.g., data scientists and researchers, to glean useful insights from datasets while ensuring that the privacy of individuals cannot be compromised. The core idea of differential privacy is that a query to a database should yield approximately the same result whether any individual person's data is included in the database or not. This is achieved by adding noise to the data or the query results, which helps to obscure the contributions of individual data points.

Since Dwork *et al.* [35] first introduced the definition of *differential privacy* (DP), DP has been extended to various scopes, such as numerical data collection [33, 134], set-value data collection [23, 136], key-value data collection [156], high-dimensional data [30], graph analysis [122], time series data release [154], private learning [46,168], federated matrix factorization [82], data mining [64], local differential privacy [5, 135, 147, 148], database query [14, 41], markov model [144] and benchmark [30, 31, 113]. Relevant to our thesis, we follow the common practice to implement Gaussian mechanism [36] to perturb model parameters. Besides, Rényi Differential Privacy (RDP) [92] allows us to more accurately estimate the cumulative privacy loss of the whole training process.

2.2.2 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a fundamental optimization algorithm widely used in machine learning and deep learning for training a wide array of models. It is especially popular for its efficiency in dealing with large datasets and high-dimensional optimization problems. SGD was first introduced by Herbert *et al.* [107], and applied for training deep learning models [109]. The development of SGD has seen several significant improvements over the years. Xavier *et al.* [52] and Yoshua [11] optimized

deep neural networks using SGD. Momentum, a critical concept to accelerate SGD, was emphasized by Llya *et al.* [123]. Diederik *et al.* [71] proposed Adam, a variant of SGD that adaptively adjusts the learning rate for each parameter. Sergey *et al.* [65] introduced Batch Normalization, a technique to reduce the internal covariate shift in deep networks. Yang *et al.* [158] and Zhang *et al.* [162] further proposed large-batch training and lookahead optimizer, respectively. These advancements have pushed the boundaries of SGD, enabling efficient training of increasingly complex deep learning models [139, 145, 146, 161]. Without loss of generality, we follow the common practice of existing works and implement SGD without momentum to better demonstrate the efficiency of our strategy.

2.2.3 Differentially Private Stochastic Gradient Descent (DP-SGD)

As a privacy-preserving technique for training various models, DP-SGD is an adaptation of the traditional SGD algorithm to incorporate differential privacy guarantees. This is crucial in applications where data confidentiality and user privacy are concerns, such as in medical or financial data processing. The basic idea is adding DP noise to gradients during the training process. Chaudhuri et al. [22] initially introduced a DP-SGD algorithm for empirical risk minimization. Abadi et al. [1] were one of the first to introduce DP-SGD into deep learning. Afterwards, DP-SGD has been rapidly applied to various models, such as generative adversarial network [62], Bayesian learning [61], federated learning [164], graph neural networks [163].

As for optimizing model efficiency of DP-SGD, there are three major streams. First, gradient clipping can help to reduce the noise scale while still following DP framework. For example, adaptive gradient clipping [24, 142, 164], which adaptively bounds the sensitivity of the DP noise, can trade the clipped information for noise reduction. Second, we can amplify the privacy bounds to save privacy budgets, such as Rényi

Differential Privacy [55]. Last, more efficient SGD algorithms, such as DP-Adam [125], can be introduced to DP-SGD so as to improve the training efficiency.

However, existing works still cling to numerical perturbation, and there is no work investigating whether the numerical DP scheme is optimal for the geometric SGD in various applications. In this thesis, we instead fill in this gap **by proposing a new DP perturbation scheme**, which exclusively preserves directions of gradients so as to improve model efficiency. As no previous works carry out optimization from this perspective, **our thesis is therefore only parallel to vanilla DP-SGD while orthogonal to all existing works**.

2.2.4 Inference Attacks (IA)

As mentioned before, DP has been introduced into SGD for defending various inference attacks, which are composed of four mainstreams, i.e., membership inference [110, 118, 119], model inversion [19, 44, 45], attribute inference [3, 91, 165], and model stealing [100, 128, 132]. The first three are designed to infer information from the target model while the last one aims to steal model parameters.

Among the four, membership inference attack, being the most basic one, is usually considered as the signal for the "leaky" model [87]. In particular, MIA aims to determine whether a specific data record was used in training a machine learning model. An adversary conducting a MIA uses access to the model (often via querying it) to infer whether specific data points were part of the model's training dataset. This type of attack poses significant privacy risks, especially when sensitive data is involved.

As one effective solution to the defense of MIA, DP [13, 96] provides a robust framework to protect against MIA by ensuring that the output of a computation is less sensitive to any single individual's data. This is achieved through mechanisms that limit information leakage, promote generalization, and introduce randomness, thereby

making it challenging for attackers to deduce the presence of specific individual data in the training set.

In this thesis, we conduct extensive experiments to evaluate the defense of our proposed strategy against real attacks, compared with vanilla DP-SGD.

2.3 Crossovers of Private Learning and Statistics Estimation

In this section, while works of LDP are previously reviewed, we continue to review related works from two aspects: federated learning and the crossover works between LDP and federated learning. On this basis, we also introduce the base of this work.

2.3.1 Federated Learning

Federated learning [72, 150], as a decentralized machine learning paradigm, allows participants to collectively train a global model without data transfer. Until now, a large body of FL works have been proposed, including efficiency improvement [57,112], federated generative adversarial network [4], medical institution collaboration [116, 117], user profiling [58], benchmark design [60], and knowledge transfer [59].

Without loss of generality, we adopt the basic algorithm of FL, a.k.a, FedAvg [90] for both analysis and experiments in this paper. As for privacy protection, we follow the common practice of existing works and adapt LDP to DP-SGD (LDP-SGD) [1]. The seminal work on FL convergence analysis [80] provides us with a theoretical foundation to analyze the impact of LDP noise on FL.

2.3.2 Integrating LDP with FL (Federated LDP-SGD)

As DP can defend federated learning against various inference attacks, their integration has led to flourishing outcomes in recent years. For example, Triastcyn *et al.* [129] consider federated reinforcement learning under DP noise. Ruan *et al.* [108] provide a practical federated DP-SGD framework under secure multi-party computation protocols. Geyer *et al.* [50] study the client-level federated DP-SGD. Seif *et al.* [114] apply federated LDP-SGD to wireless systems.

Relevant to our work are those on utility analysis and enhancement. For convergence analysis, Wei *et al.* [141] investigate FL convergence under LDP noise while Kim *et al.* [70] propose similar analysis under LDP noise. For utility enhancement, Liu *et al.* [86] optimize federated machine learning under LDP with top-k selection.

However, existing works have serious limitations. First, they are confined to Gaussian mechanism and thus lack of generality. Second, since these works are unable to accurately model the noisy training process, their analysis can only provide extremely loose bounds and thus make trivial conclusions, e.g., there is a trade-off between privacy budget and convergence rate. Third, their optimization is only effective under a large privacy budget, but leads to controversial results when applied to real world, where the privacy budget is usually limited.

Up to now, even under the relaxed LDP setting, there is no existing work that provides non-trivial analysis on federated LDP-SGD, let alone a general optimizer for it. Our thesis fits in this niche by accurately modeling the noisy training process as well as enhancing the utility of federated LDP-SGD.

Chapter 3

Preliminary

In this chapter, we introduce backgrounds of private data mining, and formulate problems to be addressed. In specific, we present , respectively.

3.1 Basic Concepts

3.1.1 Differential Privacy

Differential Privacy (DP) is a mathematical framework that quantifies the privacy preservation. Formally, (ϵ, δ)-*DP* is defined as follows:

Definition 1. ((ϵ,δ)-DP). *A randomized algorithm $\mathcal{M} : D \to R$ satisfies (ϵ,δ)-DP if for all datasets D and D' differing on a single element, and for all subsets S of R, the following inequality always holds:*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{M}(D') \in S] + \delta. \tag{3.1}$$

In essence, DP guarantees that given any outcome of $\mathcal{M}$, it is unlikely for any third party to infer the original record with high confidence. Privacy budget ϵ controls the level of preservation. Namely, a lower ϵ means stricter privacy preservation and

thus poorer efficiency, and vice versa. δ determines the probability of not satisfying ϵ preservation.

To determine the noise scale for DP, we measure the maximum change of $\mathcal{M}$ in terms of L_2-norm as:

Definition 2. *(L_2-sensitivity). The L_2-sensitivity of $\mathcal{M}$ is:*

$$\Delta\mathcal{M} = \max_{\|D-D'\|_1=1} \|\mathcal{M}(D) - \mathcal{M}(D')\|_2. \tag{3.2}$$

While tradition DP only provides a uniform privacy guarantee for all data, Rényi DP (RDP) otherwise allows varying privacy guarantees depending on the data distribution and can therefore provide tighter privacy bounds for DP-SGD.

Definition 3. *(*Rényi DP*). Rényi DP (RDP) is a generalization of (ϵ, δ)-DP that uses Rényi divergence as a distance metric. The Rényi divergence of order α between two distributions P and Q is defined as:*

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x\sim P} \left[\left(\frac{P(x)}{Q(x)} \right)^{\alpha-1} \right]. \tag{3.3}$$

A model satisfies (α, ϵ)-RDP if

$$\begin{aligned} & D_\alpha(\mathcal{M}(D)||\mathcal{M}(D')) \\ =& \frac{1}{\alpha - 1} \log \mathbb{E}_{x\sim\mathcal{M}(D)} \left[\left(\frac{\Pr[\mathcal{M}(D) = x]}{\Pr[\mathcal{M}(D') = x]} \right)^{\alpha-1} \right] \leq \epsilon. \end{aligned} \tag{3.4}$$

It can be proved [106] that an (α, ϵ)-RDP guarantee is equivalent to an $\left(\epsilon + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)$-DP guarantee for any $\delta \in (0, 1)$. As $\frac{\log(1/\delta)}{\alpha-1} > 0$ always holds, RDP provides tighter privacy bounds than (ϵ, δ)-DP.

Through out the thesis, we follow the common practice of existing works [1, 46] and use Gaussian mechanism [36] for differentially private tasks.

Gaussian Mechanism. The perturbed value of Gaussian mechanism is $g^* = g + Gau(0, 2\Delta\mathcal{M} \ln \frac{1.25}{\delta}/\epsilon^2)$, where Gau denotes a random variable that follows Gaussian

distribution with probability density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}). \tag{3.5}$$

Referring to the standard deviation of $Gau(0, 2\ln\frac{1.25}{\delta}/\epsilon^2)$ as **the noise multiplier** σ, **the noise scale** of Gaussian mechanism is $\Delta\mathcal{M}\sigma$ [36].

3.1.2 Local Differential Privacy

While DP preserves records of a dataset, local differential privacy (LDP) maintains each user's information. In LDP, let n denote the number of users and tuple $\boldsymbol{t}_i\,(1 \leq i \leq n)$ denote the i-th user's private data. To ensure privacy, each tuple $\boldsymbol{t}_i$ is locally perturbed into $\boldsymbol{t}_i^*$ by a certain perturbation mechanism $\mathcal{M}$. Afterwards, only perturbed tuples $\{\boldsymbol{t}_i^* | 1 \leq i \leq n\}$ are sent to the data collector. Given the privacy budget $\epsilon > 0$ which indicates the privacy protection level, ϵ-*local differential privacy* is formally defined as follows:

Definition 4. (ϵ-local differential privacy) *A randomized perturbation mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy if and only if for any pair of tuples $\boldsymbol{t}_i, \boldsymbol{t}_j$, the following inequality always holds:*

$$\frac{\Pr\left(\mathcal{M}\left(\boldsymbol{t}_i\right) = \boldsymbol{t}^*\right)}{\Pr\left(\mathcal{M}\left(\boldsymbol{t}_j\right) = \boldsymbol{t}^*\right)} \leq \exp\left(\epsilon\right) \tag{3.6}$$

In essence, LDP guarantees that given prior knowledge $\boldsymbol{t}^*$, it is unlikely for the data collector to identify the data source with high confidence. Privacy budget ϵ controls the trade-off between privacy protection level and utility. Lower privacy budget means stricter privacy preservation and therefore poorer utility. We then introduce three representative LDP mechanisms.

Laplace mechanism. As a classic LDP mechanism, the advantage of *Laplace mechanism* [35] is its simplicity. Given a one-dimensional value t_i in the range of $[-1, 1]$, the perturbed value is $t_i^* = t_i + Lap(2/\epsilon)$, where $Lap(\lambda)$ denotes a random variable that

follows Laplace distribution with probability density function $f(x) = \frac{1}{2\lambda}\exp(-\frac{|x|}{\lambda})$. Note that the variance of $Lap(\lambda)$ is $2\lambda^2$ [74]. To extend it to high-dimensional values, each dimension is perturbed independently with a random variable $Lap(2m/\epsilon)$ to guarantee ϵ-LDP. Since Laplace noise has zero mean, the data collector only needs to average all received tuples to achieve an unbiased mean estimation.

Generally, Laplace mechanism represents a class of LDP mechanisms [35, 49, 120], where the noise added to the original value ranges from negative to positive infinity. In this work, they are referred to as "unbounded mechanisms".

Piecewise mechanism. In one-dimensional Piecewise mechanism [134], the perturbed value t^* of an original value $t \in [-1, 1]$ follows the distribution below:

$$\Pr(t^*) = \begin{cases} \frac{e^\epsilon - e^{\epsilon/2}}{2e^{\epsilon/2}+2} & t^* \in [l(t), r(t)] \\ \frac{1-e^{-\epsilon/2}}{2e^{\epsilon/2}+2} & t^* \in [-Q, l(t)) \cup (r(t), Q] \end{cases}, \tag{3.7}$$

where

$$\begin{aligned} Q &= \frac{e^\epsilon + e^{\epsilon/2}}{e^\epsilon - e^{\epsilon/2}} \\ l(t) &= \frac{Q+1}{2}t - \frac{Q-1}{2} \\ r(t) &= l(t) + Q - 1 \end{aligned} \tag{3.8}$$

In high-dimensional space, similar to Laplace mechanism, each reporting dimension independently carries out ϵ/m-LDP. In contrast to Laplace mechanism, Piecewise mechanism perturbs the original value into a bounded domain $[-Q, Q]$, so such mechanisms are referred to as "bounded mechanisms".

Square wave mechanism. This is yet another "bounded" LDP mechanism that improves Piecewise with more concentrated perturbation [83]. In its one-dimensional form, for any original value $t \in [0, 1]$, the perturbed value $t^* \in [-b, b+1]$ follows the distribution as below:

$$\Pr(t^*) = \begin{cases} \frac{e^\epsilon}{2be^\epsilon+1} & \text{if } |t - t^*| < b \\ \frac{1}{2be^\epsilon+1} & \text{otherwise} \end{cases}, \tag{3.9}$$

where $b = \frac{\epsilon e^{\epsilon} - e^{\epsilon} + 1}{2e^{\epsilon}(e^{\epsilon} - 1 - \epsilon)}$. The ϵ-LDP is ensured by two probabilities $\frac{e^{\epsilon}}{2be^{\epsilon}+1} / \frac{1}{2be^{\epsilon}+1} = e^{\epsilon}$. Similar to Piecewise mechanism, in high-dimensional space, each reporting dimension carries out ϵ/m-LDP perturbation.

3.1.3 Differentially Private Stochastic Gradient Descent (DP-SGD)

SGD (stochastic gradient descent) is one of the most widely used optimization techniques in machine learning [15]. Let D be the private dataset, and $\boldsymbol{w}$ denote the model parameters (a.k.a the training model). Given $S \subseteq D$ and $S = \{s_1, s_2, ..., s_{(B-1)}, s_B\}$ (B denoting the number of data in S), the objective $F(\boldsymbol{w})$ can be formulated as:

$$F(\boldsymbol{w}; S) = \frac{1}{B} \sum_{j=1}^{B} l(\boldsymbol{w}; s_j). \tag{3.10}$$

where $l(\boldsymbol{w}; s_j)$ is the loss function trained on one subset data s_j to optimize $\boldsymbol{w}$.

To optimize this task, we follow the common practice of existing works and use mini-batch stochastic gradient descent (SGD) [77]. Given the total number of iterations T, $\boldsymbol{w}_t = \left(\boldsymbol{w}_{t1}, \boldsymbol{w}_{t2}, ..., \boldsymbol{w}_{t(d-1)}, \boldsymbol{w}_{td}\right)$ $(0 \leq t \leq T-1)$ denotes a d-dimensional model weight derived from the t-th iteration (where $t = 0$ is the initiate state). While using η to denote the learning rate, we have the gradient $\boldsymbol{g}_t$ of the t-th iteration:

$$\boldsymbol{g}_t = \nabla F(\boldsymbol{w}_t; S) = \frac{1}{B} \sum_{j=1}^{B} \nabla l(\boldsymbol{w}; s_j) = \frac{1}{B} \sum_{j=1}^{B} \boldsymbol{g}_{tj}. \tag{3.11}$$

where $\nabla l = \left(\frac{\partial l}{\partial \boldsymbol{w}_1}, \frac{\partial l}{\partial \boldsymbol{w}_2}, ..., \frac{\partial l}{\partial \boldsymbol{w}_{d-1}}, \frac{\partial l}{\partial \boldsymbol{w}_d}\right)$, and respective gradients $\left\{\boldsymbol{g}_{tj} | 1 \leq j \leq B\right\}$ are derived from respective data $\{s_j | 1 \leq j \leq B\}$ of the batch. The t-th iteration updates the model weight $\boldsymbol{w}_{t+1}$ as: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$.

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t \tag{3.12}$$

By tuning the batch size B, the analysis on this optimization technique also applies to its variants. For example, if $B = |D|$, it is equivalent to the batch gradient descent

[16]; if $B = 1$, it is equivalent to the stochastic gradient descent [15]. Throughout this thesis, we abbreviate mini-batch stochastic gradient descent and its variants collectively as SGD.

SGD is known to have an intrinsic problem of gradient explosion [101]. It often occurs when the gradients become very large during backpropagation, and causes the model to converge rather slowly. As the most effective solution to this problem, gradient clipping [101] is also considered in this work. Let $\|\boldsymbol{g}\|$ denote the L_2-norm of a d-dimensional vector $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_{d-1}, \boldsymbol{g}_d)$, i.e., $\|\boldsymbol{g}\| = \sqrt{\sum_{z=1}^{d} \boldsymbol{g}_z^2}$. Assume that G is the maximum L_2-norm value of all possible gradients for any weight $\boldsymbol{w}$ derived from any subset S, i.e., $G = \sup_{\boldsymbol{w} \in \mathbb{R}^d, S \in D} \mathbb{E}\left[\|\boldsymbol{g}\|\right]$. Then each gradient $\boldsymbol{g}$ is clipped by a clipping threshold $C \in (0, G]$. Formally, the clipped gradient $\tilde{\boldsymbol{g}}$ is:

$$\tilde{\boldsymbol{g}} = \frac{\boldsymbol{g}}{\max\left\{1, \|\boldsymbol{g}\| / C\right\}}. \tag{3.13}$$

Another advantage of clipping is to reduce the sensitivity of a gradient, which therefore decreases the noise addition in DP-SGD. The most recent state-of-the-art work proposes AUTO-S [17] for automatic clipping, which conducts clipping as follows:

$$\tilde{\boldsymbol{g}} = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\| + 0.01}. \tag{3.14}$$

Applying Equation 3.13 to Equation 3.11, we derive the clipped gradient from the t-th iteration as:

$$\tilde{\boldsymbol{g}}_t = \frac{1}{B} \sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{tj}. \tag{3.15}$$

3.1.4 Federated Averaging and Optimization Techniques

This work conducts analysis on the first and perhaps the most widely used FL algorithm, FedAvg (Federated Averaging) [90]. In FedAvg, the global objective is the weighted average of local objectives. Let N denote the number of local devices, and each device possesses a local dataset $D_k (1 \leq k \leq N)$. Then, we

use $\boldsymbol{w}_k\,(1 \leq k \leq N)$ to denote the local parameters (a.k.a the local model) in k-th local device. Given $S_k \subseteq D_k$ and $S_k = \left\{s_{k1}, s_{k2}, ..., s_{k(B-1)}, s_{kB}\right\}$ (where B denotes the number of data in S_k), the local objective $F_k(\boldsymbol{w})$ can be formulated as $F_k(\boldsymbol{w}; S_k) = \frac{1}{B}\sum_{j=1}^{B} l(\boldsymbol{w}; s_{kj})$, where $l(\boldsymbol{w}; s_{kj})$ is the user-specified loss function trained on one subset data s_{kj} to optimize $\boldsymbol{w}_k$. Given that p_k is the weight of k-th device such that $p_k > 0$ and $\sum_{k=1}^{N} p_k = 1$, we consider the global optimization task: $\min_{\boldsymbol{w}} \left\{F(\boldsymbol{w}) = \sum_{k=1}^{N} p_k F_k(\boldsymbol{w}; S_k)\right\}$.

To optimize this task, we follow the common practice of existing works and use mini-batch stochastic gradient descent (SGD) [77] for further analysis and experiments. Given the total number of local iterations T, $\boldsymbol{w}_k^t = \left(\boldsymbol{w}_{k1}^t, \boldsymbol{w}_{k2}^t, ..., \boldsymbol{w}_{k(d-1)}^t, \boldsymbol{w}_{kd}^t\right)$ $(0 \leq t \leq T-1)$ denotes a d-dimensional local weight vector (a.k.a the local model) derived from the t-th local iteration at the k-th local device (where $t = 0$ is the initiate state). Besides, we use η^t to denote the learning rate in the t-th local iteration. Then we have the gradient $\boldsymbol{g}_k^t$ of the t-th local iteration: $\boldsymbol{g}_k^t = \nabla F_k(\boldsymbol{w}_k^t; S_k)$, where $\nabla F_k(\boldsymbol{w}) = \left(\frac{F_k}{\partial \boldsymbol{w}_1}, \frac{F_k}{\partial \boldsymbol{w}_2}, ..., \frac{F_k}{\partial \boldsymbol{w}_{d-1}}, \frac{F_k}{\partial \boldsymbol{w}_d}\right)$. Given $\nabla_k^t = \nabla F_k(\boldsymbol{w}_k^t; D_k)$, we have $\mathbb{E}(\boldsymbol{g}_k^t) = \mathbb{E}(F_k(\boldsymbol{w}_k^t; S_k)) = \nabla_k^t$. The t-th local iteration updates local weight $\boldsymbol{w}_k^{t+1}$ as:

$$\boldsymbol{w}_k^{t+1} = \boldsymbol{w}_k^t - \eta^t \boldsymbol{g}_k^t \tag{3.16}$$

3.2 Problem Definition

3.2.1 High-dimensional Statistics Estimation

In high-dimensional settings, each $\boldsymbol{t}_i$ consists of d numerical dimensions $\boldsymbol{t}_{i1}, \boldsymbol{t}_{i2}, \ldots, \boldsymbol{t}_{id}$. Without loss of generality, we focus on mean estimation throughout this thesis and assume that the domain of any dimension ranges from $[-1, 1]$. Unless otherwise specified, we respectively use $\mathbb{E}(\cdot)$ and $Var(\cdot)$ to denote the expectation and the variance of a random variable.

Mean Estimation. We follow a common and general approach for LDP mechanisms to support high-dimensional data [33, 98, 134, 137]. Given a total privacy budget ϵ, each user randomly reports $m(1 \leq m \leq d)$ dimensions of her perturbed data to the collector, with budget ϵ/m allocated to each dimension so that ϵ-LDP still holds. Let r_j denote the number of reports that the data collector receives in the j-th dimension, and obviously $\mathbb{E}(r_i) = \frac{nm}{d}$ because randomly reporting m out of d dimensions from n users' data is statistically equal to reporting d dimensions from $\frac{nm}{d}$ users. The data collector aggregates and estimates the mean of the j-th dimension as $\hat{\boldsymbol{\theta}}_j = \frac{1}{r_j}\sum_{i=1}^{r_j} \boldsymbol{t}^*_{ij}$, so the estimated d-dimensional mean is $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, ..., \hat{\boldsymbol{\theta}}_{d-1}, \hat{\boldsymbol{\theta}}_d)^\intercal$. Note that the original mean of users is $\bar{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{t}_i$. Our objective is for the estimated mean $\hat{\boldsymbol{\theta}}$ to be as close to the original mean $\bar{\boldsymbol{\theta}}$ as possible. Therefore, we adopt the following utility metrics which can measure their difference.

Utility Metrics. Theoretically, their difference can be measured by the *Euclidean distance*, i.e.,

$$\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 = \sqrt{\sum_{j=1}^{d} |\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j|^2} \tag{3.17}$$

Following [35, 83, 134], we adopt *mean square error* (MSE) to measure experimental error, namely, the average squared difference between estimated means and original means over all dimensions, i.e.,

$$\text{MSE}(\hat{\boldsymbol{\theta}}) = \frac{1}{d}\sum_{j=1}^{d} |\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j|^2 \tag{3.18}$$

Applying Equation 3.17 to Equation 3.18, we have $\text{MSE}(\hat{\boldsymbol{\theta}}) = \frac{1}{d}\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2^2$, which indicates that the theoretical analysis on $\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2$ can predict how MSE varies without conducting any experiment.

3.2.2 Private Learning Analysis and Enhancement

As shown in Algorithm 1, in each iteration of DP-SGD, $\boldsymbol{w}_{t+1}$ is perturbed to $\boldsymbol{w}^*_{t+1}$ by adding DP noise $\boldsymbol{n}_t$ to the sum of $\tilde{\boldsymbol{g}}_{tj}$. Let $\boldsymbol{g}^*_t$ denote the perturbed gradient. Formally,

$$\begin{aligned} \boldsymbol{g}^*_t &= \frac{1}{B}(\sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{tj} + \boldsymbol{n}_t) = \tilde{\boldsymbol{g}}_t + \boldsymbol{n}_t/B, \\ \boldsymbol{w}^*_{t+1} &= \boldsymbol{w}_t - \eta \boldsymbol{g}^*_t. \end{aligned} \tag{3.19}$$

Accordingly, the following definition establishes the measurement for model efficiency (ME). Obviously, a smaller ME means a better model efficiency.

Definition 5. *(Model Efficiency (ME)). Suppose there exists a global optima $\boldsymbol{w}^\star$, the model deficiency can be measured by the Euclidean Distance between the current model $\boldsymbol{w}^*_{t+1}$ and the optima $\boldsymbol{w}^\star$, i.e.,*

$$\textit{Model efficiency (ME)} = \left\| \boldsymbol{w}^*_{t+1} - \boldsymbol{w}^\star \right\|^2. \tag{3.20}$$

Besides, it is essential to control the overall privacy cost in the whole training process. For Laplace mechanism, only composition theorem [68] is applicable. For Gaussian mechanism, on the other hand, the moments accountant [1] is an economic method to measure the collective privacy cost.

Definition 6. *(Moments Accountant). Suppose a model is trained for T iterations with a batch size B on a dataset of size $|D|$. For any $\epsilon < \frac{c_1 B^2 T}{N^2}$, there exists such constants c_1 and c_2 that the model is (ϵ, δ)-DP for any $\delta > 0$ if we choose the noise multiplier σ to be:*

$$\sigma \geq c_2 \frac{B\sqrt{T \log(1/\delta)}}{|D|\epsilon}. \tag{3.21}$$

To accumulate the overall privacy cost of the training process, composition theorem of RDP [68] is applicable. As having to validate the optimality of GeoDP over DP on preserving the descent trend, we follow the common practice [134] and adopt mean

square error (MSE) to measure the error on perturbed directions. In general, a larger MSE means a larger perturbation.

Definition 7. *(Mean Square Error (MSE)). Considering the perturbed directions* $\{\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, ..., \boldsymbol{\theta}_{m-1}^*, \boldsymbol{\theta}_m^*\}$ *and the original directions* $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{m-1}, \boldsymbol{\theta}_m\}$ *of* m *gradients, MSE of perturbed directions is defined as follows:*

$$MSE(\boldsymbol{\theta}^*) = \frac{1}{m}\sum_{i=1}^{m} \|\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_i\|_2^2. \tag{3.22}$$

The problem in this work is to investigate the impact of DP noise $\boldsymbol{n}_t$ on the SGD efficiency, i.e., $\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2$, and further optimize the model efficiency by reducing the noise on the direction of a gradient, i.e., reducing MSE($\boldsymbol{\theta}^*$).

Algorithm 1 DP-SGD

Input: Batch size B, noise multiplier σ, clipping threshold C, learning rate η, total number of iterations T.

Output: Trained model parameters $\boldsymbol{w}_T$.

1: Initialize a model with parameters $\boldsymbol{w}_0$.
2: **for** $t = 0$ to $T - 1$ **do**
3: Derive the average clipped gradient $\tilde{\boldsymbol{g}}_t$ with respect to the sampled subset $S \in D$ and the clipping threshold C.
4: Add noise $\boldsymbol{n}_t$ drawn from a zero-mean Gaussian distribution with standard deviation $\sigma C\boldsymbol{I}$ to $\tilde{\boldsymbol{g}}_t$, i.e., $\boldsymbol{g}_t^* = \tilde{\boldsymbol{g}}_t + \boldsymbol{n}_t/B$, where $\boldsymbol{n}_t$ is jointly determined by both σ and C.
5: Update $\boldsymbol{w}_{t+1}^*$ by taking a step in the direction of the noisy gradient, i.e., $\boldsymbol{w}_{t+1}^* = \boldsymbol{w}_t - \eta \boldsymbol{g}_t^*$.
6: **end for**

3.2.3 Model Efficiency of Federated LDP-SGD

For crossovers works between private learning and statistics estimation, without loss of generality, we consider FedAvg under SGD with LDP noise, as indicated in Algorithm 2. Before global aggregation, $\boldsymbol{w}_k^{t+1}$ is perturbed to $\boldsymbol{w}_k^{t+1*}$ by adding LDP noise $\boldsymbol{n}_k^t$ to $\tilde{\boldsymbol{g}}_k^t$, before being sent to the central server. Let $\boldsymbol{g}_k^{t*}$ denote the perturbed local gradient. Formally, $\boldsymbol{g}_k^{t*} = \tilde{\boldsymbol{g}}_k^t + \boldsymbol{n}_k^t$. On receiving local perturbed models from all devices, the central server aggregates them to obtain the current global model $\boldsymbol{w}^{t+1*}$. Suppose there exists a global optima $\boldsymbol{w}^\star$, the convergence can be measured by the Euclidean Distance between the current global model $\boldsymbol{w}^{t+1*}$ and the optima $\boldsymbol{w}^\star$, i.e., $\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\|^2$. Given the convergence of federated SGD $\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\|^2$, their difference $\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\|^2$ reflects the model efficiency of federated LDP-SGD. In specific, the smaller difference means better efficiency, and vice versa.

The problem in this work is to first theoretically model this difference in terms of various LDP mechanisms. Then we identify the negative impact of LDP perturbation and accordingly optimize model efficiency.

Algorithm 2 Federated LDP-SGD

Input: Initiate global model $\boldsymbol{w}^0$, LDP noise $\boldsymbol{n}_k^t$, training rounds T/E, local dataset D_k, learning rate η^t.

Output: The updated global model $\boldsymbol{w}^{t+1*}$.

Initialize global model parameters $\boldsymbol{w}^0$.

for each round $T/E = 1, 2, \ldots$ **do**

    for each local device k **do**

        Send current global updates $\boldsymbol{w}^t$ to device k.

        Device k trains a local model with SGD on its dataset D_k using $\boldsymbol{w}_k^t$, and obtains the local perturbed model $\boldsymbol{w}_k^{t+1*} = \boldsymbol{w}_k^t - \eta^t(\tilde{\boldsymbol{g}} + \boldsymbol{n}_k^t)$.

        Send local perturbed update $\boldsymbol{w}_k^{t+1*}$ to the server.

    end for

    Server aggregates the local models using a weighted average: $\boldsymbol{w}^{t+1*} = \sum_{k=1}^{N} p_k \boldsymbol{w}_k^{t+1*}$.

end for

Chapter 4

Analyzing and Enhancing LDP Mechanisms in High-dimensional Space

While frequently-used notations are listed in Table 4.1, this chapter introduce our general toolbox *LDPTube* (see Figure 4.1 for overview), which generally analyzes and enhances utilities of various LDP mechanisms in high-dimensional space. In recent years, with growing number of IoT and smart devices, a huge amount of data becomes more accessible than ever [7, 51, 84, 88, 104]. Thanks to the advancement of modern machine learning and deep learning technologies, service providers and researchers nowadays can get insight into users' behavior and intention with simple clicks. However, together with the prevalence of these technologies emerge privacy concerns during the collection of sensitive data about users. To balance data utility and privacy disclosure, an effective and highly recognized solution is local differential privacy (LDP) [32, 105, 151], where the data collector only collects perturbed data from users.

Nevertheless, existing LDP mechanisms mostly focus on low-dimensional data, mainly

because the statistics estimated in high-dimensional space have low accuracy. As users only authorize a limited privacy budget to the collector, the allocated privacy budget in each dimension is diluted as the number of dimensions increases, which leads to more information loss and poorer statistics accuracy. Although much attention has been paid to develop less perturbed LDP mechanisms for multi-dimensional data [49, 83, 120, 134], they are still not applicable in high-dimensional space.

In this chapter, we first propose an analytical framework that generalizes LDP mechanisms and derives their utilities in high-dimensional space, namely the probability density function of the deviation between the estimated mean and the true mean. The framework can serve as a benchmark to compare the utilities of various LDP mechanisms without conducting any experiment. Furthermore, our analysis shows the sub-optimality of the naive aggregation method of all LDP mechanisms — the utility deterioration is attributed to the overwhelming noise caused by diluted privacy budget in high-dimensional space. As such, our second contribution in this chapter is a one-off, non-iterative re-calibration protocol *HDR4ME* (acronym for High-Dimensional Re-calibration for Mean Estimation). Through regularization and proximal gradient descent, this protocol re-calibrates the aggregated mean obtained from any LDP mechanism by suppressing the overwhelming noise and thus enhances its utility. Without any change on the LDP mechanism itself, *HDR4ME* can be used as a general optimizer of existing LDP mechanisms in high-dimensional space.

The rest of this chapter is organized as follows. We introduce the analytical framework for high-dimensional LDP mechanisms in Section 4.1 and propose our mean estimation protocol in Section 4.2. Extensive experimental results are demonstrated in Section 4.3, and summaries are made in Section 4.4.

Table 4.1: Notations

Symbol	**Meaning**
n	number of users
d	number of dimensions
$\mathcal{M}$	perturbation mechanism
$\boldsymbol{t}_i$	user's private tuple
$\boldsymbol{t}_i^*$	user's perturbed tuple
m	number of sampled dimensions
r	aggregator's received reports
$\bar{\boldsymbol{\theta}}$	original mean
$\hat{\boldsymbol{\theta}}$	estimated mean
$\boldsymbol{\theta^*}$	enhanced mean
$\mathcal{L}$	loss function
$\mathcal{R}$	regularizer
$\boldsymbol{\lambda}^*$	regularization weight

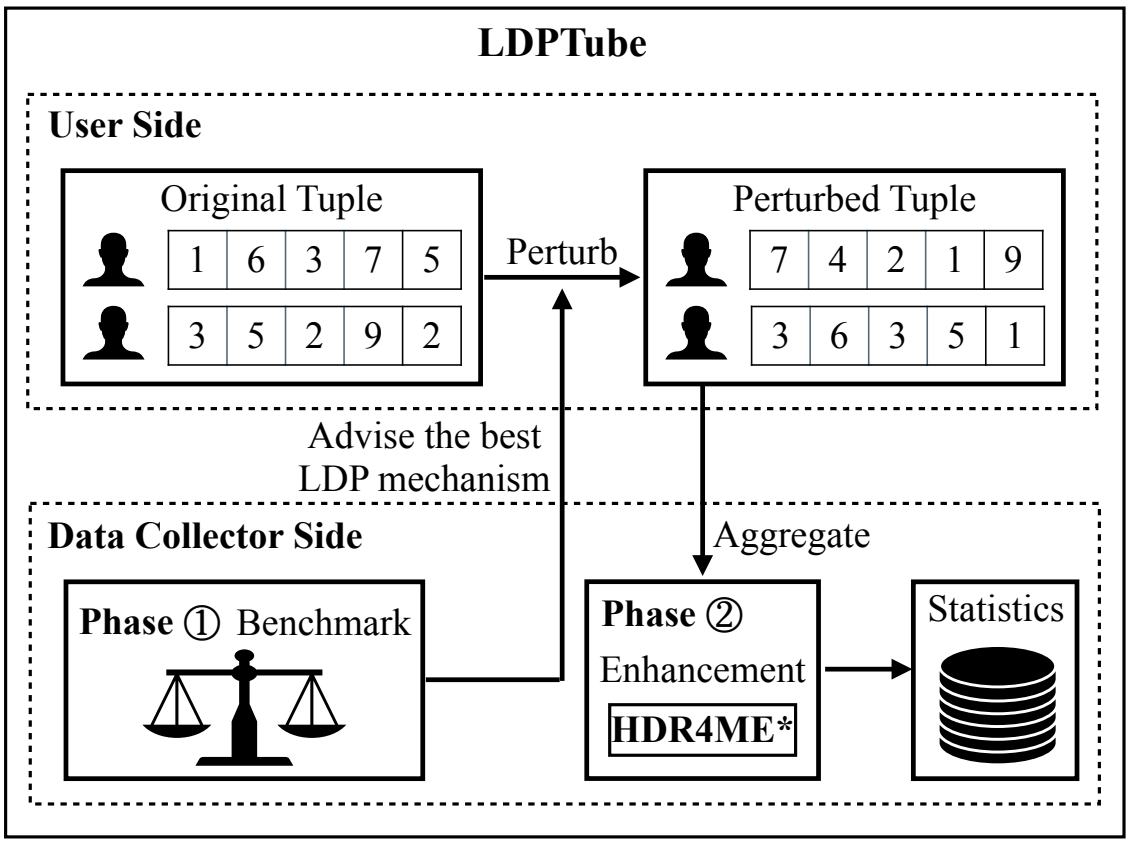


Figure 4.1: Overview of LDPTube.

4.1 A General Analytical Framework

In this section, we present our general framework for high-dimensional LDP mechanisms. As aforementioned, we first use a boolean *Bound* to denote whether the perturbation of a certain LDP mechanism $\mathcal{M}$ has a finite "boundary" B. Then a d-dimensional LDP mechanism with privacy budget ϵ is generalized as follows:

- *Perturbation:* Each user has a private tuple $\boldsymbol{t}_i\,(1 \leq i \leq n)$, among which m dimensional values are perturbed and reported. For each dimension $j\,(1 \leq j \leq d)$, the mechanism obfuscates $\boldsymbol{t}_{ij}$ to $\boldsymbol{t}^*_{ij}$ with budget ϵ/m. If $Bound(\mathcal{M}) = 1$, the perturbed tuple satisfies $\boldsymbol{t}^*_i = \mathcal{M}(\boldsymbol{t}_i) \in [-B, B]^d$, where B is a both positive and finite value. Otherwise, the perturbed tuple satisfies $\boldsymbol{t}^*_i = \mathcal{M}(\boldsymbol{t}_i) = \boldsymbol{t}_i + \boldsymbol{N}_i$, where $\boldsymbol{N}_i$ denotes a random tuple from $\mathbb{R}^d$.

- *Calibration:* In each dimension j, the data collector receives r_j reports, where $r = \mathbb{E}(r_j) = \frac{nm}{d}$. Letting $\boldsymbol{\delta}_{ij}$ denote the bias of $\mathbb{E}(\boldsymbol{t}^*_{ij})$, we have $\boldsymbol{\delta}_{ij} = \mathbb{E}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij})$. Accordingly, the collector calibrates the perturbed values by $\boldsymbol{\delta}_{ij}$. Note that $\boldsymbol{\delta}_{ij} = 0$ carries out unbiased estimation.

- *Aggregation:* For mean estimation in j-th dimension, the mechanism averages all calibrated values to obtain the estimated mean $\hat{\boldsymbol{\theta}}_j = \frac{1}{r_j}\sum_{i=1}^{r_j} \boldsymbol{t}^*_{ij}$.

Under this framework, we analyze the utility of high-dimensional LDP mechanisms based on the theoretical distance between the original mean $\bar{\boldsymbol{\theta}}$ and the estimated mean $\hat{\boldsymbol{\theta}}$ using *Lindeberg–Lévy Central Limit Theorem* (CLT) [43, 115]. Since each dimension is independently perturbed, we first model the deviation $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in one dimension.

Lemma 1. *For any $\mathcal{M}$ and ϵ/m, $Var(\boldsymbol{t}^*_{ij})$ and $\boldsymbol{\delta}_{ij}$ are deterministic if $Bound(\mathcal{M}) = 0$ while correlated to $\boldsymbol{t}_{ij}$ if $Bound(\mathcal{M}) = 1$.*

Proof. If $Bound(\mathcal{M}) = 0$, $Var(\boldsymbol{t}_{ij}^*) = Var(\boldsymbol{t}_{ij} + \boldsymbol{N}_{ij}) = Var(\boldsymbol{t}_{ij}) + Var(\boldsymbol{N}_{ij}) = Var(\boldsymbol{N}_{ij})$ while $\boldsymbol{\delta}_{ij} = \mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) = \mathbb{E}(\boldsymbol{N}_{ij})$. Since $\boldsymbol{N}_{ij}$ follows one perturbation, both $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$ are independent of $\boldsymbol{t}_{ij}$. If $Bound(\mathcal{M}) = 1$, different $\boldsymbol{t}_{ij}$ correspond with different perturbations. Otherwise, $\boldsymbol{t}_{ij}^*$ would be totally independent from $\boldsymbol{t}_{ij}$. In this case, $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$ depend on $\boldsymbol{t}_{ij}$. □

Lemma 1 derives some common properties on $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$. Given $\mathcal{M}$ and ϵ/m, $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$ are certain functions of ϵ/m if $Bound(\mathcal{M}) = 0$. Otherwise, they are certain functions of both ϵ/m and $\boldsymbol{t}_{ij}$. As long as the perturbation is known, we are able to provide $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$ considering $Var(\boldsymbol{t}_{ij}^*) = \mathbb{E}({\boldsymbol{t}_{ij}^*}^2) - \mathbb{E}^2(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij} = \mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij})$. For further utility analysis, we assume that $Var(\boldsymbol{t}_{ij}^*)$ and $\boldsymbol{\delta}_{ij}$ are already provided given certain $\mathcal{M}$ and ϵ/m. Because $\lim_{r_j \to \infty} \left(\frac{1}{r_j} \sum_{i=1}^{r_j} \boldsymbol{t}_{ij} - \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{t}_{ij} \right) = 0$, the deviation $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ can be simplified if $r_j \to \infty$:

$$
\begin{aligned}
&\lim_{r_j \to \infty} \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j \\
&= \lim_{r_j \to \infty} \left(\frac{1}{r_j} \sum_{i=1}^{r_j} \boldsymbol{t}_{ij}^* - \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{t}_{ij} \right) \\
&= \lim_{r_j \to \infty} \left(\frac{1}{r_j} \sum_{i=1}^{r_j} \boldsymbol{t}_{ij}^* - \frac{1}{r_j} \sum_{i=1}^{r_j} \boldsymbol{t}_{ij} + \frac{1}{r_j} \sum_{i=1}^{r_j} \boldsymbol{t}_{ij} - \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{t}_{ij} \right) \\
&= \lim_{r_j \to \infty} \frac{1}{r_j} \sum_{i=1}^{r_j} \left(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} \right).
\end{aligned}
\tag{4.1}
$$

Suppose that X is a random variable following standard normal distribution $X \sim \mathcal{N}(0, 1)$, and its probability density function is $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$, the following two lemmas establish the asymptotic distribution of the deviation in one dimension.

Lemma 2. $\lim_{r_j \to \infty} \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j \sim \mathcal{N}\left(\mathbb{E}(\boldsymbol{N}_{ij}), \frac{Var(\boldsymbol{N}_{ij})}{r_j} \right)$, *if* $Bound(\mathcal{M}) = 0$.

Proof. For $\forall j$, $\left\{ \boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} | 1 \leq i \leq r_j \right\}$ are independent and identically distributed (i.i.d.) random variables because $\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} = \boldsymbol{N}_{ij}$. According to *Lindeberg–Lévy Central Limit*

Theorem [43, 115], the following probability holds:

$$\begin{aligned}
&\lim_{r_j\to\infty} \Pr\left(\frac{\frac{1}{r_j}\sum_{i=1}^{r_j}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - \mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij})}{\sqrt{Var(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij})/r_j}} \le X\right) \\
&= \lim_{r\to\infty} \Pr\left(\frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \mathbb{E}(\boldsymbol{N}_{ij})}{\sqrt{Var(\boldsymbol{N}_{ij})/r_j}} \le X\right) \\
&= \int_{-\infty}^{X} \phi(x)dx.
\end{aligned} \tag{4.2}$$

Thus, $\lim_{r_j\to\infty} \frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \mathbb{E}(\boldsymbol{N}_{ij})}{\sqrt{Var(\boldsymbol{N}_{ij})/r_j}}$ follows standard normal distribution $\mathcal{N}(0,1)$, by which our claim is proven. □

In Lemma 1, both $Var(\boldsymbol{t}_{ij}^*) = Var(\boldsymbol{N}_{ij})$ and $\boldsymbol{\delta}_{ij} = \mathbb{E}(\boldsymbol{N}_{ij})$ are deterministic if $Bound(\mathcal{M}) = 0$. On this basis, we could approximate $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ using a specific Gaussian distribution $\mathcal{N}(\mathbb{E}(\boldsymbol{N}_{ij}), Var(\boldsymbol{N}_{ij})/r_j)$ if $Bound(\mathcal{M}) = 0$.

However, it is rather challenging if $Bound(\mathcal{M}) = 1$. Lemma 1 proves that different original values follow different perturbations if $Bound(\mathcal{M}) = 1$. Consequently, $\left\{\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} | 1 \le i \le r_j\right\}$ are probably not identically distributed, which does not satisfy the prerequisite of CLT [43, 115].

Nevertheless, we are still able to use one Gaussian distribution to approximate the summation of elements in any particular subset $\left\{\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}\right\}$, where all original data have the same value, and therefore CLT can be applied. Note that $\left\{\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} | 1 \le i \le r_j\right\}$ can be divided into several particular subsets by different original values. Let $\{v_j | 1 \le j \le d\}$ denote numbers of different original values in each dimension, $\left\{\boldsymbol{p}_{zj} | \sum_{z=1}^{v_j} \boldsymbol{p}_{zj} = 1\right\}$ denote their corresponding probabilities. As regards original data following continuous distribution, we discretize them with sampling. The following lemma establishes the asymptotic distribution of the deviation in one dimension if $Bound(\mathcal{M}) = 1$, where we assume $\left\{\boldsymbol{t}_{ij}^* | 1 \le i \le r_j, 1 \le j \le d\right\}$ is in ascending order in each dimension.

Lemma 3. $\lim_{r_j\to\infty} \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j \sim \mathcal{N}\left(\mathbb{E}(\boldsymbol{\delta}_{ij}), \frac{\mathbb{E}(Var(\boldsymbol{t}_{ij}^*))}{r_j}\right)$, *where* $\mathbb{E}(\boldsymbol{\delta}_{ij}) = \sum_{z=1}^{v_j} \boldsymbol{p}_{zj} \boldsymbol{\delta}_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j}$ *and* $\mathbb{E}(Var(\boldsymbol{t}_{ij}^*)) = \sum_{z=1}^{v_j} \boldsymbol{p}_{zj} Var\left(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j}\right)$, *if* $Bound(\mathcal{M}) = 1$.

Proof. For $1 \leq c \leq v_j$, the original data in $\left\{\boldsymbol{t}_{ij} | r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj} < i \leq r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}\right\}$ share the same value. Therefore, $\left\{\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} | r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj} < i \leq r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}\right\}$ are i.i.d. random variables. According to *Lindeberg–Lévy Central Limit Theorem* [43,115], the following probability holds if r_j approaches ∞:

$$\begin{aligned}
&\Pr\left(\frac{\sum_{i=r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj}+1}^{r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}} \left(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \mathbb{E}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij})\right)}{\sqrt{Var(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) r_j \boldsymbol{p}_{cj}}} \leq X\right) \\
&= \Pr\left(\frac{\sum_{i=r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj}+1}^{r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}} \left(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}\right)}{\sqrt{Var(\boldsymbol{t}^*_{ij}) r_j \boldsymbol{p}_{cj}}} \leq X\right) \\
&= \int_{-\infty}^{X} \phi(x) dx.
\end{aligned} \tag{4.3}$$

Therefore, $\frac{\sum_{i=r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj}+1}^{r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}} \left(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}\right)}{\sqrt{Var(\boldsymbol{t}^*_{ij}) r_j \boldsymbol{p}_{cj}}}$ approximately follows standard normal distribution $\mathcal{N}(0,1)$. Next, we use *Mathematical Induction* to complete the proof.

For $c = 1$, Equation 4.3 establishes $\sum_{i=1}^{r_j \boldsymbol{p}_{1j}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) \sim \mathcal{N}(0, r_j \boldsymbol{p}_{1j} Var(\boldsymbol{t}^*_{(r_j \boldsymbol{p}_{1j})j}))$.

Suppose that $\sum_{i=1}^{r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) \sim \mathcal{N}(0, \sum_{z=1}^{c} r_j \boldsymbol{p}_{zj} Var(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j}))$ holds for $1 < c < v_j$, we have the following for $r_j \to \infty$:

$$\begin{aligned}
\sum_{i=1}^{r_j \sum_{z=1}^{c+1} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) = &\sum_{i=1}^{r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) \\
&+ \sum_{i=r_j \sum_{z=1}^{c} \boldsymbol{p}_{zj}+1}^{r_j \sum_{z=1}^{c+1} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}).
\end{aligned} \tag{4.4}$$

According to Equation 4.3, $\sum_{i=r_j \sum_{z=1}^{c-1} \boldsymbol{p}_{zj}+1}^{r_j \sum_{z=1}^{c+1} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij})$ follows $\mathcal{N} \sim (0, r_j \boldsymbol{p}_{(c+1)j} Var(\boldsymbol{t}^*_{ij}))$. Given $Y \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Z \sim \mathcal{N}(\mu_2, \sigma_2^2)$, $Y + Z \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$ [78]. Therefore, we prove that for $1 < c < v_j$ and $r_j \to \infty$,

$$\begin{aligned}
&\sum_{i=1}^{r_j \sum_{z=1}^{c+1} \boldsymbol{p}_{zj}} (\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) \\
&\sim \mathcal{N}\left(0, \sum_{z=1}^{c+1} r_j \boldsymbol{p}_{zj} Var(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r \boldsymbol{p}_{oj})j})\right).
\end{aligned} \tag{4.5}$$

Letting $c = v_j - 1$, if r_j approaches ∞, we have:

$$\begin{aligned}
&\Pr\left(\frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \mathbb{E}(\boldsymbol{\delta}_{ij})}{\sqrt{\sum_{z=1}^{v_j} \boldsymbol{p}_{zj} Var(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j})/r_j}} \leq X\right) \\
=&\Pr\left(\frac{\sum_{i=1}^{r_j}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) - \sum_{z=1}^{v_j} r_j \boldsymbol{p}_{zj} \boldsymbol{\delta}_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j}}{\sqrt{\sum_{z=1}^{v_j} r_j \boldsymbol{p}_{zj} Var(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j})}} \leq X\right) \\
=&\Pr\left(\frac{\sum_{i=1}^{r_j}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij})}{\sqrt{\sum_{z=1}^{v_j} r_j \boldsymbol{p}_{zj} Var(\boldsymbol{t}^*_{(\sum_{o=1}^{z} r_j \boldsymbol{p}_{oj})j})}} \leq X\right) \\
=&\int_{-\infty}^{X} \phi(x)dx.
\end{aligned} \tag{4.6}$$

by which our claim is proven. □

Note that $\mathbb{E}(\boldsymbol{\delta}_{ij})$ and $\mathbb{E}(Var(\boldsymbol{t}^*_{ij}))$ computes the expectations of $\boldsymbol{\delta}_{ij}$ and $Var(\boldsymbol{t}^*_{ij})$ in terms of $\boldsymbol{t}^*_{ij}$. In general, Lemma 2 and Lemma 3 establish that no matter how the original data is distributed, $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ always approximates a normal distribution. However, its variance is split into two cases. If $Bound(\mathcal{M}) = 0$, it is only decided by the distribution of perturbation; otherwise, it is collectively decided by distributions of both perturbation and original data. As such, given a certain dataset and a budget, we can model how $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ varies in terms of any mechanism.

What if multiple or even high dimensions? Note that each dimension is independently perturbed with privacy budget ϵ/m. As each dimension of the deviation approximates a one-dimensional normal distribution, we can model the deviation $\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}$ with one multivariate normal distribution. Following Lemma 2 or Lemma 3, for $1 \leq j \leq d$, $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ approximates a normal distribution whose probability density function is $f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}_j} \exp(-\frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \boldsymbol{\delta}_j)^2}{2\boldsymbol{\sigma}_j^2})$. Then the following theorem models the deviation in high-dimensional space.

Theorem 1. *For any high-dimensional LDP mechanism, the probability density func-*

tion (pdf) of $\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}$ *is:*

$$\lim_{r\to\infty} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}) = \frac{1}{(\sqrt{2\pi})^d \prod_{j=1}^{d} \boldsymbol{\sigma}_j} \exp\left(-\sum_{j=1}^{d} \frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \boldsymbol{\delta}_j)^2}{2\boldsymbol{\sigma}_j^2}\right). \tag{4.7}$$

Proof. Since each dimension is perturbed independently, we have:

$$\begin{aligned} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}) &= \prod_{j=1}^{d} f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \boldsymbol{\delta}_j)^2}{2\sigma_j^2}) \\ &= \frac{1}{(\sqrt{2\pi})^d \prod_{j=1}^{d} \sigma_j} \exp(-\sum_{j=1}^{d} \frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \boldsymbol{\delta}_j)^2}{2\sigma_j^2}). \end{aligned} \tag{4.8}$$

□

As this pdf models how $\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}$ varies in high-dimensional space, we can accommodate almost all utility metrics for comparisons, including the supremum of the deviation. To benchmark different LDP mechanisms, intuitively the smallest supremum of the deviation $\sup \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2$ should have the best utility. However, due to the randomness in LDP mechanisms, the absolute supremum can be infinity. As such, the data collector can manually specify the supremum of deviation she wants to tolerate, and then calculate the corresponding probability for that supremum to hold using this pdf. Let $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_d)^\intercal = \left(\sup\left|\hat{\boldsymbol{\theta}}_1 - \bar{\boldsymbol{\theta}}_1\right|, ..., \sup\left|\hat{\boldsymbol{\theta}}_d - \bar{\boldsymbol{\theta}}_d\right|\right)^\intercal$ denotes the supremum and $S = \left\{\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}} \in \mathbb{R}^d : \forall j, \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| \leq \boldsymbol{\xi}_j\right\}$ denotes the subspace bounded by the supremum, then the integral of the pdf $\int_S f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\mathrm{d}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$ is the probability of the deviation within the supremum. Accordingly, the LDP mechanism with the highest probability is considered the best in high-dimensional space. Note that different supremum settings can lead to different winners.

4.1.0.1 Saving The Burden of Choosing Parameters

There are three issues in the previous benchmark. First, the supremum of deviation is required as an input parameter for utility benchmark and comparison. However,

the data collector, who has no access to the original dataset, can hardly determine a precise supremum, which makes the benchmark less practical. Second, as the population may vary from time to time, the data collector has to constantly re-benchmark different LDP mechanisms in order to choose the best. Third, the baseline benchmark can only output the probability of the deviation exceeding the supremum, which is different from the prevalent utility metrics of LDP mechanisms, such as mean square error (MSE). The misalignment may cause wrong comparison result if MSE is the actual utility metric aimed by the collector.

To address these issues, in this subsection we propose a non-parametric benchmark which outputs MSE directly, as MSE is more prevalent than MAE (mean absolute error). Furthermore, it can also derive the break-even point of two LDP mechanisms in terms of population to avoid re-benchmark during user change.

Now that Theorem 1 models how $\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}$ varies in high-dimensional space, we can accommodate MSE as the utility metric for estimated mean. In each dimension j, we have:

$$\begin{aligned}
\mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right)^2\right\} =& \mathbb{E}\left\{\left[\left(\hat{\boldsymbol{\theta}}_j - \mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right)\right) + \left(\mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right) - \bar{\boldsymbol{\theta}}_j\right)\right]^2\right\} \\
=& \mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}}_j - \mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right)\right)^2\right\} \\
&+ \mathbb{E}\left\{\left(\mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right) - \bar{\boldsymbol{\theta}}_j\right)^2\right\} \\
&+ 2\mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}}_j - \mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right)\right)\left(\mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right) - \bar{\boldsymbol{\theta}}_j\right)\right\} \\
=& \mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}}_j - \mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right)\right)^2\right\} \\
&+ \mathbb{E}\left\{\left(\mathbb{E}\left(\hat{\boldsymbol{\theta}}_j\right) - \bar{\boldsymbol{\theta}}_j\right)^2\right\} \\
=& Var(\hat{\boldsymbol{\theta}}_j) + \boldsymbol{\delta}_j^2.
\end{aligned} \tag{4.9}$$

In particular, $\mathbb{E}\left\{(\hat{\boldsymbol{\theta}}_j - \mathbb{E}(\hat{\boldsymbol{\theta}}_j))(\mathbb{E}(\hat{\boldsymbol{\theta}}_j) - \bar{\boldsymbol{\theta}}_j)\right\} = (\mathbb{E}(\hat{\boldsymbol{\theta}}_j) - \mathbb{E}(\hat{\boldsymbol{\theta}}_j))(\mathbb{E}(\hat{\boldsymbol{\theta}}_j) - \bar{\boldsymbol{\theta}}_j) = 0$. Recall that the deviation always approximates a Gaussian distribution in our framework. That is to say, the variance of the estimated mean equivalently approximates that of

a Gaussian distribution. Namely, $Var(\hat{\boldsymbol{\theta}}_j) = Var(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \boldsymbol{\sigma}_j^2$. On this basis, we obtain the MSE of the estimated mean in high-dimensional space:

$$\begin{aligned} \mathrm{MSE}(\hat{\boldsymbol{\theta}}) =& \mathbb{E}\left\{\frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right)^2\right\} \\ =& \frac{\sum_{j=1}^{d}\mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right)^2\right\}}{d} \\ =& \frac{\sum_{j=1}^{d}\left(Var(\hat{\boldsymbol{\theta}}_j) + \boldsymbol{\delta}_j^2\right)}{d} \\ =& \frac{\sum_{j=1}^{d}\left(\boldsymbol{\sigma}_j^2 + \boldsymbol{\delta}_j^2\right)}{d}. \end{aligned} \tag{4.10}$$

Equation 4.10 harmoniously unifies two common metrics for utility analysis, namely, estimation bias $\boldsymbol{\delta}$ [35, 83, 134], and variance of perturbed values $\boldsymbol{\sigma}$ [35, 134]. This indicates that MSE, besides being a popular metric in real-life applications, can also serve as a more comprehensive metric than bias or variance alone. Taking Square wave and Piecewise mechanisms for example, the former is biased with a lower variance of perturbed values [134] while the latter is unbiased with a larger variance [83]. Although a lower variance in Square wave means better utility at the first glance, its biased estimation can accumulate errors with increasing number of users.

4.1.0.2 Computational Saving Under Population Variation in Practice

Besides relieving the burden of choosing a supremum parameter, this non-parametric benchmark has another unprecedented advantage. In face of population variation, it can directly derive the break-even point where two mechanisms perform the same whereas the previous benchmark has to re-benchmark for each user change.

Recall there are three variables in Equation 3.18, i.e., the number of dimensions, the bias and the variance of the Gaussian distribution. When deriving the MSE of an LDP mechanism with fixed privacy budget, the number of dimensions and the bias are fixed, and only the variance is changed according to user population. Then

according to either Lemma 2 or Lemma 3, the numerator in the variance of Gaussian approximation, i.e., the expectation of the perturbed value's variance, is also fixed due to fixed LDP mechanism and privacy budget. However, its denominator, i.e., the number of reports, is proportional to the user population. As such, our benchmark can be simplified as a function where the input is the number of users and the output is MSE. By comparing the simplified utility functions of two LDP mechanisms, we can derive the break-even point of user population, below or above which one better than the other and vice versa. Based on this break-even point, instead of conducting re-benchmark from time to time, the data collector could easily tell which mechanism is better.

In what follows, we use Square wave and Piecewise mechanisms as an example to derive this break-even point. Following Equation 3.18, the MSE of Square wave is

$$\frac{\sum_{j=1}^{d}\left(\mathbb{E}(Var(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j^2\right)}{d}, \tag{4.11}$$

while that of Piecewise is

$$\frac{\sum_{j=1}^{d}\left(\mathbb{E}(Var'(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j'^2\right)}{d}. \tag{4.12}$$

Canceling the invariant term d in both MSEs, we only need to compare $\sum_{j=1}^{d}\left(\mathbb{E}(Var(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j^2\right)$ and $\sum_{j=1}^{d}\left(\mathbb{E}(Var'(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j'^2\right)$. Since each dimension is independently perturbed, we further derive their MSEs in each dimension and make them equal as the equation below:

$$\mathbb{E}(Var(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j^2=\mathbb{E}(Var'(\boldsymbol{t}_{ij}^{*}))/r_j+\boldsymbol{\delta}_j'^2. \tag{4.13}$$

By solving this equation, we derive the report threshold as:

$$r_j'=\mathbb{E}\left(Var'(\boldsymbol{t}_{ij}^{*})-Var(\boldsymbol{t}_{ij}^{*})\right)/\left(\boldsymbol{\delta}_j^2-\boldsymbol{\delta}_j'^2\right). \tag{4.14}$$

Considering the conversion between the number of reports and the number of users (i.e., $r=\frac{nm}{d}$), we finally obtain the population threshold $n_j'=\frac{r_j'd}{m}$. That is, in j-th

dimension, Square wave is the winner if it has a break-even point below n'_j. Otherwise, the winner is Piecewise. The population threshold can be negative, which means that Square wave is always better than Piecewise in such setting. The psuedo-code is

Algorithm 3 Theoretical Non-parametric Benchmark for Population Variation

Input: MSE of LDP mechanism A: $\frac{\sum_{j=1}^{d}\left(\mathbb{E}(Var(\boldsymbol{t}^*_{ij}))/r_j+\boldsymbol{\delta}_j^2\right)}{d}$, MSE of LDP mechanism B: $\frac{\sum_{j=1}^{d}\left(\mathbb{E}(Var'(\boldsymbol{t}^*_{ij}))/r_j+\boldsymbol{\delta}_j'^2\right)}{d}$.

Output: The threshold population n'_j in j-th dimension, the comparison result in high-dimensional space.

1: **for** $j = 1$ to d **do**

2: compute break-even point

$$n'_j = \frac{\mathbb{E}\left(Var'(\boldsymbol{t}^*_{ij}) - Var(\boldsymbol{t}^*_{ij})\right)}{\left(\boldsymbol{\delta}_j^2 - \boldsymbol{\delta}_j'^2\right)} \times \frac{d}{m};$$

3: **if** $n'_j \leq 0$ **then**

4: B is always the winner in j-th dimension;

5: **else if** $n \leq n'_j$ **then**

6: A is the winner in j-th dimension;

7: **else**

8: B is the winner in j-th dimension;

9: **end if**

10: **end for**

listed in Algorithm 3, which facilitates the data collector to choose the better LDP mechanism in each dimension. In the next subsection, we provide a comprehensive case study to demonstrate how to benchmark Square wave and Piecewise with our benchmark.

4.1.1 Approximation Error of Theorem 1

Our analytical framework is based on one assumption that the data collector receives sufficiently large number of reports from users. Otherwise, the *central limit theorem* provides an asymptotic approximation of the deviation. In order to find the gap between the approximated deviation and the true one, we study the approximation error of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in terms of the number of reports r_j. Suppose the true *pdf* of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ is $\bar{f}_j$, its corresponding cumulative distribution function (*cdf*) would be $\bar{F}_j(x) = \int_{-\infty}^{x} \bar{f}_j(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)\mathrm{d}(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)$. According to Lemma 2 or Lemma 3, the approximated *pdf* of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ is $\hat{f}_j(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}_j} \exp(-\frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \boldsymbol{\delta}_j)^2}{2\boldsymbol{\sigma}_j^2})$, and its corresponding *cdf* is $\hat{F}_j(x) = \int_{-\infty}^{x} \hat{f}_j(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)\mathrm{d}(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)$. Then we have:

Theorem 2. *For any LDP mechanism, the true cdf $\bar{F}_j(x)$ and the approximated cdf $\hat{F}_j(x)$ of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ differ by no more than $\frac{0.33554(\rho + 0.415(r_j\boldsymbol{\sigma}_j)^3)}{r_j^{7/2}\boldsymbol{\sigma}_j^3}$, where we have $\rho = \mathbb{E}\left(\left|\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}\right|^3\right)$.*

Proof. As necessary prerequisites, $\mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) = 0$, and Lemma 2 and Lemma 3 prove that $\mathbb{E}((\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij})^2) = \mathbb{E}(Var(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}) + \mathbb{E}^2(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij})) = \mathbb{E}(Var(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij})) = \mathbb{E}(Var(\boldsymbol{t}_{ij}^*)) = (r_j\boldsymbol{\sigma}_j)^2$. Besides, we have to prove $\rho < \infty$. If $Bound(\mathcal{M}) = 1$, it surely establishes because $\boldsymbol{t}_{ij}^*$, $\boldsymbol{t}_{ij}$ and $\boldsymbol{\delta}_{ij}$ are all finite values in this case. If $Bound(\mathcal{M}) = 0$, we can prove that *Laplace mechanism* satisfies this term. Note that $\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij} = \boldsymbol{N}_{ij}$. Therefore, we have:

$$
\begin{aligned}
\rho &= \mathbb{E}\left(\left|\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij} - \boldsymbol{\delta}_{ij}\right|^3\right) = \int_{-\infty}^{\infty} |x|^3 \, Lap(\lambda = 2m/\epsilon)\mathrm{d}x \\
&= \frac{1}{\lambda}\int_0^{\infty} x^3 \exp(-\frac{x}{\lambda})\mathrm{d}x = -\int_0^{\infty} x^3 d(\exp(-\frac{x}{\lambda})) \\
&= 3\int_0^{\infty} x^2 \exp(-\frac{x}{\lambda})dx - x^3 \exp(-\frac{x}{\lambda})|_0^{\infty} = 3\int_0^{\infty} x^2 \exp(-\frac{x}{\lambda})dx \\
&= \frac{3\lambda}{2}\int_{-\infty}^{\infty} \frac{x^2}{\lambda} \exp(-\frac{|x|}{\lambda})dx \\
&= \frac{3\lambda}{2}\mathbb{E}(x^2) = \frac{3\lambda}{2} 2\lambda^2 = 3\lambda^3 = \frac{24m^3}{\epsilon^3} < \infty.
\end{aligned}
\tag{4.15}
$$

Further, we have

$$\begin{aligned}\bar{F}_j(\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j) &= \int_{-\infty}^{\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j} \bar{f}_j(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)d(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)\\ &= \int_{-\infty}^{x} \bar{f}_j(\frac{\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j-\boldsymbol{\delta}_j}{\boldsymbol{\sigma}_j})d(\frac{\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j-\boldsymbol{\delta}_j}{\boldsymbol{\sigma}_j}).\end{aligned} \tag{4.16}$$

While

$$\begin{aligned}\hat{F}_j(\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j) &= \int_{-\infty}^{\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j} \hat{f}_j(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)d(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)\\ &= \int_{-\infty}^{\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j} \frac{1}{\boldsymbol{\sigma}_j}\phi(\frac{\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j-\boldsymbol{\delta}_j}{\boldsymbol{\sigma}_j})d(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)\\ &= \int_{-\infty}^{\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j} \phi(\frac{\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j-\boldsymbol{\delta}_j}{\boldsymbol{\sigma}_j})d(\frac{\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j-\boldsymbol{\delta}_j}{\boldsymbol{\sigma}_j})\\ &= \int_{-\infty}^{x} \phi(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j)d(\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j).\end{aligned} \tag{4.17}$$

As such, *Berry–Esseen theorem* [73] establishes:

$$\sup_{x\in\mathbb{R}}\left|\bar{F}_j(x)-\hat{F}_j(x)\right| = \sup_{x\in\mathbb{R}}\left|\bar{F}_j(\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j)-\hat{F}_j(\boldsymbol{\sigma}_j x+\boldsymbol{\delta}_j)\right| \leq \frac{0.33554(\rho+0.415(r_j\boldsymbol{\sigma}_j)^3)}{r_j^{7/2}\boldsymbol{\sigma}_j^3}. \tag{4.18}$$

□

In particular, $\sup_{x\in\mathbb{R}}\left|\bar{F}_j(x)-\hat{F}_j(x)\right| \to 0$ if $r\to\infty$. Namely, the approximated distribution converges to the real one as long as r is sufficiently large. According to Lemma 2 and Lemma 3, the value of $r_j\boldsymbol{\sigma}_j$ is irrelevant to r_j. Thus, $r_j\boldsymbol{\sigma}_j$ can be taken as a fixed value, which implies that the speed of convergence rate in our framework is at least on the order of $\frac{r_j^3}{r_j^{7/2}}=\frac{1}{\sqrt{r_j}}$. That is to say, the approximation error is still tolerable even if the number of reports is insufficient. We take *Laplace mechanism* for example, where $\rho=3\lambda^3$ in Equation 4.15 and $r_j\boldsymbol{\sigma}_j=\sqrt{Var(\boldsymbol{t}_{ij}^*)}=\sqrt{(Var(Lap(\lambda)))}=\sqrt{2}\lambda$. Suppose the data collector only receives $r_j=1000$ reports, the approximation error between the true cdf and the approximated cdf of $\hat{\boldsymbol{\theta}}_j-\bar{\boldsymbol{\theta}}_j$ is no more than $\frac{0.33554(\rho+0.415(r_j\boldsymbol{\sigma}_j)^3)}{r_j^{7/2}\boldsymbol{\sigma}_j^3}=\frac{0.33554\times(3\times\lambda^3+0.415\times2\times\sqrt{2}\times\lambda^3)}{2\times\sqrt{2}\times\lambda^3\times\sqrt{r_j}}\approx 1.57\%$.

4.1.2 A Case Study: How to Benchmark Piecewise Mechanism and Square Wave Mechanism in High-Dimensional Space?

Since each dimension is perturbed equivalently in high-dimensional space, we study how to benchmark these two mechanisms in any single dimension. Suppose an original dataset with $d = 100$ dimensions and $n = 10000$ users, there are $v = 10$ different original values $\{0.1, 0.2, 0.3, ..., 0.8, 0.9, 1.0\}$ in each dimension. For simplicity, we presume that the corresponding probability of each value in each dimension is $p = 10\%$. For each user, she reports $m = 100$ dimensions of her data to the data collector. As such, the data collector receives $r = \frac{nm}{d} = 10000$ reports. Given the collective privacy budget $\epsilon = 0.1$, each dimension is allocated $\epsilon/m = 0.001$ privacy budget. Next, we demonstrate how to obtain the pdf in Theorem 1 for each LDP mechanism. For Piecewise mechanism, we first obtain the variance of $\boldsymbol{t}^*_{ij}$:

$$\begin{aligned} Var(\boldsymbol{t}^*_{ij}) =& \mathbb{E}({\boldsymbol{t}^*_{ij}}^2) - \mathbb{E}^2(\boldsymbol{t}^*_{ij}) \\ =& \int_{-Q}^{l(\boldsymbol{t}^*_{ij})} \frac{(1-e^{-\epsilon/2m})x^2}{2e^{\epsilon/2m}+2}\mathrm{d}x \\ &+ \int_{l(\boldsymbol{t}^*_{ij})}^{r(\boldsymbol{t}^*_{ij})} \frac{(e^{\epsilon/m}-e^{\epsilon/2m})x^2}{2e^{\epsilon/2m}+2}\mathrm{d}x \\ &+ \int_{r(\boldsymbol{t}^*_{ij})}^{Q} \frac{(1-e^{-\epsilon/2m})x^2}{2e^{\epsilon/2m}+2}\mathrm{d}x \\ =& \frac{\boldsymbol{t}^*_{ij}}{e^{\epsilon/2m}-1} + \frac{e^{\epsilon/2m+3}}{3(e^{\epsilon/2m}-1)^2}. \end{aligned} \tag{4.19}$$

We then derive the variance $\boldsymbol{\sigma}^2_j$ of Gaussian distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ according to Lemma 3:

$$\begin{aligned} \boldsymbol{\sigma}^2_j &= \frac{\sum_{z=1}^{v} pVar\left(\boldsymbol{t}^*_{(\sum_{o=1}^{z} rp)j}\right)}{r} \\ &= \frac{\frac{10\%\times(0.1+0.2+...+1.0)}{e^{0.001/2}-1} + \frac{e^{0.001/2+3}}{3(e^{0.001/2}-1)^2}}{10000} \\ &= 533.210. \end{aligned} \tag{4.20}$$

Due to unbiased estimation, we can derive the pdf of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in Piecewise mechanism by applying $d = 1$, $\boldsymbol{\sigma}_j^2 = 533.210$, and $\boldsymbol{\delta}_j = 0$ to Equation 4.7:

$$f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \frac{1}{57.900} \exp\left(-\frac{(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)^2}{1066.420}\right). \tag{4.21}$$

For the Square wave mechanism, we have the bias of $\mathbb{E}(\boldsymbol{t}_{ij}^*)$:

$$\begin{aligned}
\boldsymbol{\delta}_{ij} =& \mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) \\
=& \int_{-b}^{\boldsymbol{t}_{ij}-b} \frac{x}{2be^{\epsilon/m}+1}\mathrm{d}x + \int_{\boldsymbol{t}_{ij}-b}^{\boldsymbol{t}_{ij}+b} \frac{xe^{\epsilon/m}}{2be^{\epsilon/m}+1}\mathrm{d}x \\
&+ \int_{\boldsymbol{t}_{ij}+b}^{1+b} \frac{x}{2be^{\epsilon/m}+1}\mathrm{d}x - \boldsymbol{t}_{ij} \\
=& \frac{2b(e^{\epsilon/m}-1)\boldsymbol{t}_{ij}}{2be^{\epsilon/m}+1} + \frac{1+2b}{2(2be^{\epsilon/m}+1)} - \boldsymbol{t}_{ij}.
\end{aligned} \tag{4.22}$$

and the variance of $\boldsymbol{t}_{ij}^*$:

$$\begin{aligned}
Var(\boldsymbol{t}_{ij}^*) =& \mathbb{E}({\boldsymbol{t}_{ij}^*}^2) - \mathbb{E}^2(\boldsymbol{t}_{ij}^*) \\
=& \int_{-b}^{\boldsymbol{t}_{ij}-b} \frac{x^2}{2be^{\epsilon/m}+1}\mathrm{d}x + \int_{\boldsymbol{t}_{ij}-b}^{\boldsymbol{t}_{ij}+b} \frac{x^2e^{\epsilon/m}}{2be^{\epsilon/m}+1}\mathrm{d}x \\
&+ \int_{\boldsymbol{t}_{ij}+b}^{1+b} \frac{x^2}{2be^{\epsilon/m}+1}\mathrm{d}x - (\boldsymbol{t}_{ij} + \boldsymbol{\delta}_{ij})^2 \\
=& \frac{b^2}{3} + \frac{(2b+1)(b+1-3\boldsymbol{t}_{ij}^2)}{3(2be^{\epsilon/m}+1)} - \boldsymbol{\delta}_{ij}^2 - 2\boldsymbol{\delta}_{ij}\boldsymbol{t}_{ij}.
\end{aligned} \tag{4.23}$$

We then derive the bias $\boldsymbol{\delta}_j$ and the variance $\boldsymbol{\sigma}_j^2$ of the Gaussian distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ according to Lemma 3:

$$\begin{aligned}
\boldsymbol{\delta}_j &= \sum_{z=1}^{v} p\boldsymbol{\delta}_{ij} = -0.049, \\
\boldsymbol{\sigma}_j^2 &= \frac{\sum_{z=1}^{v} pVar\left(\boldsymbol{t}^*_{(\sum_{o=1}^{z} rp)j}\right)}{r} = 3.365 \times 10^{-5}.
\end{aligned} \tag{4.24}$$

Finally, according to Theorem 1, we can derive the pdf of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in the Square wave mechanism by applying Equation 4.24 and $d = 1$ to Equation 4.7:

$$f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) = \frac{1}{0.015} \exp\left(-\frac{10^5(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j + 0.049)^2}{6.730}\right). \tag{4.25}$$

Now that we have derived the pdf of $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in both LDP mechanisms, its integral $\int_{-\boldsymbol{\xi}_j}^{\boldsymbol{\xi}_j} f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)\mathrm{d}(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)$ is the probability that the deviation in j-th dimension is still within the supremum $\boldsymbol{\xi}_j = \sup|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j|$. The higher probability the better the LDP mechanism. We vary $\boldsymbol{\xi}_j$ from 0.001 to 0.1 and show the resulted probabilities in Table 4.2. Piecewise mechanism is better than Square wave mechanism for smaller supremums (e.g., $0.001, 0.01$), which is mainly because Piecewise is an unbiased estimation while Square wave is not. However, if the supremum becomes larger (e.g., $0.05, 0.1$), in other words, if the collector can tolerate larger deviation, the Square wave mechanism is far better than the Piecewise mechanism because the variance of Gaussian distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in the former is much smaller than that in the latter. That is to say, whether Piecewise or Square wave should be chosen depend on her tolerance of supremum $\boldsymbol{\xi}_j$.

Table 4.2: Probabilities for the supremum to hold in one dimension

$\boldsymbol{\xi}_j$	0.001	0.01	0.05	0.1
Piecewise	3.46×10^{-5}	3.46×10^{-4}	0.002	0.004
Square	2.12×10^{-16}	2.62×10^{-11}	0.644	1.000

Suppose a data collector is curious about the worst utility with confidence level $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_d)^\intercal$. Note that each entry of $\boldsymbol{\alpha}$ represents her confidence level in each dimension. Therefore, we have $\sup\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| = \arg_{\boldsymbol{\xi}_j \in \mathbb{R}^+}\left\{\boldsymbol{\alpha}_j = \int_{-(\boldsymbol{\xi}_j - \boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j}^{(\boldsymbol{\xi}_j - \boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j} \phi(x)dx\right\}$. Then we obtain $\sup\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 = \sqrt{\sum_{j=1}^{d} \sup\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|^2}$ in terms of each LDP mechanism. As mentioned before, **one with the smallest supremum may be benchmarked to have the most excellent general utility in high-dimensional space**.

Besides, the data collector may implement the optimal combination of LDP mech-

anisms following our framework. That is to say, she may choose the most excellent LDP mechanism in each dimension so that the collective utility is optimal. Thus, the key point is how to benchmark LDP mechanisms in each dimension. Given the limit of estimation error $\boldsymbol{\xi}_j$ or the confidence level $\boldsymbol{\alpha}_j$ in j-th dimension, $\int_{-(\boldsymbol{\xi}_j-\boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j}^{(\boldsymbol{\xi}_j-\boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j} \phi(x)dx$ is the probability where $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ may not exceed the limit while $\arg_{\boldsymbol{\xi}_j \in \mathbb{R}^+} \left\{ \boldsymbol{\alpha}_j = \int_{-(\boldsymbol{\xi}_j-\boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j}^{(\boldsymbol{\xi}_j-\boldsymbol{\delta}_j)/\boldsymbol{\sigma}_j} \phi(x)dx \right\}$ is the supremum of $\left| \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j \right|$ under such a confidence level. Namely, **the candidate with the highest probability or the smallest supremum can be selected as the most suitable one in this dimension**. By repeating the above steps in each dimension can the data collector finally obtain the optimal combination of LDP mechanisms in high-dimensional space.

Also, non-parametric form of benchmark can also make the same judgment. For the Square wave, we have the bias of $\mathbb{E}(\boldsymbol{t}_{ij}^*)$:

$$\begin{aligned}
\boldsymbol{\delta}_{ij} =& \mathbb{E}(\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) \\
=& \int_{-b}^{\boldsymbol{t}_{ij}-b} \frac{x}{2be^{\epsilon/m}+1}dx + \int_{\boldsymbol{t}_{ij}-b}^{\boldsymbol{t}_{ij}+b} \frac{xe^{\epsilon/m}}{2be^{\epsilon/m}+1}dx \\
&+ \int_{\boldsymbol{t}_{ij}+b}^{1+b} \frac{x}{2be^{\epsilon/m}+1}dx - \boldsymbol{t}_{ij} \\
=& \frac{2b(e^{\epsilon/m}-1)\boldsymbol{t}_{ij}}{2be^{\epsilon/m}+1} + \frac{1+2b}{2(2be^{\epsilon/m}+1)} - \boldsymbol{t}_{ij}.
\end{aligned} \tag{4.26}$$

and the variance of $\boldsymbol{t}_{ij}^*$:

$$\begin{aligned}
Var(\boldsymbol{t}_{ij}^*) =& \mathbb{E}({\boldsymbol{t}_{ij}^*}^2) - \mathbb{E}^2(\boldsymbol{t}_{ij}^*) \\
=& \int_{-b}^{\boldsymbol{t}_{ij}-b} \frac{x^2}{2be^{\epsilon/m}+1}dx + \int_{\boldsymbol{t}_{ij}-b}^{\boldsymbol{t}_{ij}+b} \frac{x^2e^{\epsilon/m}}{2be^{\epsilon/m}+1}dx \\
&+ \int_{\boldsymbol{t}_{ij}+b}^{1+b} \frac{x^2}{2be^{\epsilon/m}+1}dx - (\boldsymbol{t}_{ij}+\boldsymbol{\delta}_{ij})^2 \\
=& \frac{b^2}{3} + \frac{(2b+1)(b+1-3\boldsymbol{t}_{ij}^2)}{3(2be^{\epsilon/m}+1)} - \boldsymbol{\delta}_{ij}^2 - 2\boldsymbol{\delta}_{ij}\boldsymbol{t}_{ij}.
\end{aligned} \tag{4.27}$$

Further, we have the expectation of the variance of t_{ij}^*:

$$\mathbb{E}(Var(\boldsymbol{t}_{ij}^*)) = \sum_{z=1}^{v} pVar\left(\boldsymbol{t}^*_{(\sum_{o=1}^{z} rp)j}\right) = 1.483 \times 10^{-1}. \tag{4.28}$$

We then derive the bias $\boldsymbol{\delta}_j$ and the variance $\boldsymbol{\sigma}_j^2$ of the Gaussian distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ according to Lemma 3:

$$\begin{aligned} \boldsymbol{\delta}_j &= \sum_{z=1}^{v} p\boldsymbol{\delta}_{ij} = -3.160 \times 10^{-2} \\ \boldsymbol{\sigma}_j^2 &= \frac{\mathbb{E}(Var(\boldsymbol{t}_{ij}^*))}{r} = 1.483 \times 10^{-5}. \end{aligned} \tag{4.29}$$

From Equation 3.18, we have:

$$\text{MSE}(\hat{\boldsymbol{\theta}}_j) = \boldsymbol{\sigma}_j^2 + \boldsymbol{\delta}_j^2 = 1.014 \times 10^{-3}. \tag{4.30}$$

For Piecewise, we first obtain the variance of $\boldsymbol{t}_{ij}^*$:

$$\begin{aligned} Var'(\boldsymbol{t}_{ij}^*) =& \mathbb{E}({\boldsymbol{t}_{ij}^*}^2) - \mathbb{E}^2(\boldsymbol{t}_{ij}^*) \\ =& \int_{-Q}^{l(\boldsymbol{t}_{ij}^*)} \frac{(1 - e^{-\epsilon/2m})x^2}{2e^{\epsilon/2m} + 2} dx \\ &+ \int_{l(\boldsymbol{t}_{ij}^*)}^{r(\boldsymbol{t}_{ij}^*)} \frac{(e^{\epsilon/m} - e^{\epsilon/2m})x^2}{2e^{\epsilon/2m} + 2} dx \\ &+ \int_{r(\boldsymbol{t}_{ij}^*)}^{Q} \frac{(1 - e^{-\epsilon/2m})x^2}{2e^{\epsilon/2m} + 2} dx \\ =& \frac{\boldsymbol{t}_{ij}}{e^{\epsilon/2m} - 1} + \frac{e^{\epsilon/2m+3}}{3(e^{\epsilon/2m} - 1)^2}. \end{aligned} \tag{4.31}$$

Further, we have the expectation of the variance of t_{ij}^*:

$$\begin{aligned} \mathbb{E}(Var'(\boldsymbol{t}_{ij}^*)) &= \sum_{z=1}^{v} pVar'\left(\boldsymbol{t}^*_{(\sum_{o=1}^{z} rp)j}\right) \\ &= \frac{10\% \times (0.1 + 0.2 + ... + 1.0)}{e^{1/2} - 1} + \frac{e^{1/2+3}}{3(e^{1/2} - 1)^2} \\ &= 27.08. \end{aligned} \tag{4.32}$$

We then derive the variance $\boldsymbol{\sigma}_j'^2$ of Gaussian distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ according to Lemma 3:

$$\begin{aligned} \boldsymbol{\sigma}_j'^2 &= \frac{\mathbb{E}(Var'(\boldsymbol{t}_{ij}^*))}{r} \\ &= \frac{27.08}{10000} = 2.708 \times 10^{-3}. \end{aligned} \tag{4.33}$$

Note that Piecewise is unbiased [134], which means $\boldsymbol{\delta}'_j = 0$. From Equation 3.18, we have:

$$\text{MSE}'(\hat{\boldsymbol{\theta}}_j) = \boldsymbol{\sigma}'^2_j + \boldsymbol{\delta}'^2_j = 2.708 \times 10^{-3} > \text{MSE}(\hat{\boldsymbol{\theta}}_j). \tag{4.34}$$

That is, Square wave is better than Piecewise in this scenario, and at least $n'_j = \frac{\mathbb{E}\left(Var'(\boldsymbol{t}^*_{ij}) - Var(\boldsymbol{t}^*_{ij})\right)}{\left(\boldsymbol{\delta}^2_j - \boldsymbol{\delta}'^2_j\right)} \times \frac{d}{m} = 26960$ users are required for Piecewise to outperform Square wave based on our benchmark, below which Square wave is always better.

4.1.3 Utility Analysis in Personalized Local Differential Privacy (PLDP)

In regular LDP, the privacy budget and the privacy region (a.k.a. the domain of the original tuple) are normally set by the data collector. Nevertheless, it is common that users may not be satisfied with the uniform settings and prefer to configure these settings by themselves. PLDP, as a special case of LDP, allows each user to decide her own privacy budget ϵ and privacy region τ [149]. With ϵ and τ configured by the user, PLDP guarantees that the user can enjoy ϵ-LDP in the subdomain $\tau \subseteq D_{\mathcal{M}}$ of the original domain. To provide general utility analysis for PLDP, we also adapt our framework as per users' personalized privacy preferences.

Without loss of generality, we study the utility analysis in a single dimension. Let $\mathcal{E}$ denote the available privacy budgets $\mathcal{E} = \{\epsilon_1, \epsilon_2, ..., \epsilon_{b-1}, \epsilon_b\}$ and $\mathcal{T}$ denote the privacy regions $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_{c-1}, \tau_c\}$. Each user chooses her own preferable privacy combination in this dimension and share this information with the data collector. While users perturb their original value as per personal privacy preferences, the data collector groups all users into $b \times c$ subgroups who have the same privacy combination and thus the same perturbation. As such, we can implement *central limit theorem* [43, 115] to each subgroup[1] according to Lemma 2 or Lemma 3. Given the set T_j

[1]For simplicity, here we assume that each subgroup has a sufficiently large number of users. In practice, those subgroups without enough users, e.g., fewer than 100, are ignored.

of original data and the number of reports r_j in j-th dimension, let T_{jz} and r_{jz} ($1 \leq z \leq bc$) denote the subset of original data and the number of reports in z-th subgroup, respectively. Obviously, T_j and r_j can be decomposed by $T_j = \bigcap_{z=1}^{b\times c} T_{jz}$ and $r_j = \sum_{z=1}^{b\times c} r_{jz}$, respectively. Denoted as Var_{jz} and $\boldsymbol{\delta}_{jz}$, we can then obtain the expectation of the variance of the perturbed data $\left\{\boldsymbol{t}^*_{ij} | \boldsymbol{t}_{ij} \in T_{jz}\right\}$ and the bias of the perturbed data in z-th subgroup, respectively. That is, $Var_{jz} = \mathbb{E}_{\boldsymbol{t}_{ij}\in T_{jz}}(Var(\boldsymbol{t}^*_{ij}))$ while $\boldsymbol{\delta}_{jz} = \mathbb{E}_{\boldsymbol{t}_{ij}\in T_{jz}}(\mathbb{E}(\boldsymbol{t}^*_{ij}) - \boldsymbol{t}_{ij})$. The following theorem establishes the foundation of utility analysis in terms of deviation in any dimension j.

Theorem 3. $\lim\limits_{r_j\to\infty} \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j \sim \mathcal{N}\left(\frac{\sum_{z=1}^{bc} r_{jz}\boldsymbol{\delta}_{jz}}{r_j}, \frac{\sum_{z=1}^{bc} r_{jz} Var_{jz}}{r_j^2}\right)$ *in PLDP.*

Proof. Within any subgroup, any original data $\boldsymbol{t}_{ij}$ is perturbed into a perturbed data $\boldsymbol{t}^*_{ij}$ under the same perturbation, which is the same scenario in either Lemma 2 or Lemma 3. On this basis, the following probability holds according to *Lindeberg–Lévy Central Limit Theorem* [43, 115]:

$$\begin{aligned}
&\lim_{\substack{r_j\to\infty \\ \boldsymbol{t}_{ij}\in T_{jz}}} \Pr\left(\frac{\sum(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) - r_{jz}\mathbb{E}_{\boldsymbol{t}_{ij}\in T_{jz}}(\mathbb{E}(\boldsymbol{t}^*_{ij}) - \boldsymbol{t}_{ij})}{\sqrt{r_{jz}Var_{jz}}} \leq X\right) \\
&= \lim_{\substack{r_j\to\infty \\ \boldsymbol{t}_{ij}\in T_{jz}}} \Pr\left(\frac{\sum(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) - r_{jz}\boldsymbol{\delta}_{jz}}{\sqrt{r_{jz}Var_{jz}}} \leq X\right) \\
&= \int_{-\infty}^{X} \phi(x)dx.
\end{aligned} \tag{4.35}$$

Thus, $\lim\limits_{\substack{r_j\to\infty \\ \boldsymbol{t}_{ij}\in T_{jz}}} \frac{\sum(\boldsymbol{t}^*_{ij}-\boldsymbol{t}_{ij})-r_{jz}\boldsymbol{\delta}_{jz}}{\sqrt{r_{jz}Var_{jz}}}$ follows the standard normal distribution $\mathcal{N}(0,1)$. Next, we prove the theorem by mathematical induction on z.

For $z = 1$, Equation 4.35 establishes $\sum_{\boldsymbol{t}_{ij}\in T_{jz}}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) - r_{jz}\boldsymbol{\delta}_{jz} \sim \mathcal{N}(0, r_{jz}Var_{jz})$.

Suppose that $\sum_{\boldsymbol{t}_{ij}\in\bigcap_{z=1}^{k} T_{jz}}(\boldsymbol{t}^*_{ij} - \boldsymbol{t}_{ij}) - \sum_{z=1}^{k} r_{jz}\boldsymbol{\delta}_{jz} \sim \mathcal{N}(0, \sum_{z=1}^{k} r_{jz}Var_{jz})$ holds for

$1 < k < bc$, we have the following if $r_j \to \infty$:

$$\begin{aligned}
&\sum_{\boldsymbol{t}_{ij} \in \bigcap_{z=1}^{k+1} T_{jz}} (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - \sum_{z=1}^{k+1} r_{jz} \boldsymbol{\delta}_{jz} \\
= &\sum_{\boldsymbol{t}_{ij} \in \bigcap_{z=1}^{k} T_{jz}} (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - \sum_{z=1}^{k} r_{jz} \boldsymbol{\delta}_{jz} \\
&+ \left(\sum_{\boldsymbol{t}_{ij} \in T_{j(k+1)}} (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - r_{j(k+1)} \boldsymbol{\delta}_{j(k+1)} \right).
\end{aligned} \tag{4.36}$$

According to Equation 6.9, $\sum_{\boldsymbol{t}_{ij} \in T_{j(k+1)}} (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - r_{j(k+1)} \boldsymbol{\delta}_{j(k+1)} \sim \mathcal{N}(0, r_{j(k+1)} Var_{j(k+1)})$. Given $Y \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Z \sim \mathcal{N}(\mu_2, \sigma_2^2)$, $Y + Z \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$ [78]. Therefore, we prove the following for $1 < k < bc$ and $r_j \to \infty$:

$$\begin{aligned}
&\sum_{\boldsymbol{t}_{ij} \in \bigcap_{z=1}^{k+1} T_{jz}} (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - \sum_{z=1}^{k+1} r_{jz} \boldsymbol{\delta}_{jz} \\
\sim &\mathcal{N}(0, \sum_{z=1}^{k+1} r_{jz} Var_{jz}).
\end{aligned} \tag{4.37}$$

Applying $k = bc - 1$ to Equation 4.37, we have for $r_j \to \infty$:

$$\begin{aligned}
&\lim_{\substack{r_j \to \infty \\ \boldsymbol{t}_{ij} \in \bigcap_{z=1}^{bc} T_{jz}}} \Pr \left(\frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j - \frac{\sum_{z=1}^{bc} r_{jz} \boldsymbol{\delta}_{jz}}{r_j}}{\frac{\sqrt{\sum_{z=1}^{bc} r_{jz} Var_{jz}}}{r_j}} \leq X \right) \\
= &\lim_{\substack{r_j \to \infty \\ \boldsymbol{t}_{ij} \in \bigcap_{z=1}^{bc} T_{jz}}} \Pr \left(\frac{\frac{\sum (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij})}{r_j} - \frac{\sum_{z=1}^{bc} r_{jz} \boldsymbol{\delta}_{jz}}{r_j}}{\frac{\sqrt{\sum_{z=1}^{bc} r_{jz} Var_{jz}}}{r_j}} \leq X \right) \\
= &\lim_{\substack{r_j \to \infty \\ \boldsymbol{t}_{ij} \in \bigcap_{z=1}^{bc} T_{jz}}} \Pr \left(\frac{\sum (\boldsymbol{t}_{ij}^* - \boldsymbol{t}_{ij}) - \sum_{z=1}^{bc} r_{jz} \boldsymbol{\delta}_{jz}}{\sqrt{\sum_{z=1}^{bc} r_{jz} Var_{jz}}} \leq X \right) \\
= &\int_{-\infty}^{X} \phi(x) dx.
\end{aligned} \tag{4.38}$$

which proves the theorem. □

Theorem 3 shows that the deviation in each dimension can be modeled by a Gaussian distribution. As such, the total deviation in all dimensions follows a multivariate

Gaussian distribution as in Theorem 1. Deriving the biases and the variances of the deviations from Theorem 3, respectively:

$$\begin{aligned} \boldsymbol{\delta}_j &= \frac{\sum_{z=1}^{bc} r_{jz}\boldsymbol{\delta}_{jz}}{r_j}, \\ \boldsymbol{\sigma}_j^2 &= \frac{\sum_{z=1}^{bc} r_{jz}Var_{jz}}{r_j^2}. \end{aligned} \tag{4.39}$$

Applying them into Equation 3.18, the MSE can be obtained as:

$$\begin{aligned} \boldsymbol{\delta}_j &= \frac{\sum_{j=1}^{d}\left(\boldsymbol{\sigma}_j^2 + \boldsymbol{\delta}_j^2\right)}{d} \\ &= \sum_{j=1}^{d} \frac{\sum_{z=1}^{bc} r_{jz}Var_{jz} + \left(\sum_{z=1}^{bc} r_{jz}\boldsymbol{\delta}_{jz}\right)^2}{r_j^2 d}. \end{aligned} \tag{4.40}$$

4.2 HDR4ME*: High-dimensional Re-calibration for Mean Estimation

In our analytical framework, we observe that dimensions d has significant and direct influence on the deviation. In specific, d dictates the privacy budget in each dimension, which directly affects the accuracy. In this section, we seize this opportunity to reduce the effective d in the aggregation phase to improve the accuracy. The rationale of targeting at the aggregation phase instead of the perturbation or calibration is obvious — the latter are mechanism-dependent whereas the former is universal to all LDP mechanisms. As such, our enhancement is orthogonal to all existing LDP optimizations.

In what follows, we first introduce *regularization* that can mitigate the negative influence in high dimensions. By integrating it into the aggregation, we propose a *re-calibration protocol HDR4ME** and a solver algorithm based on proximal gradient descent. Last, we extend *HDR4ME** for frequency estimation. Rigorous analysis is provided to prove its superiority over the existing one.

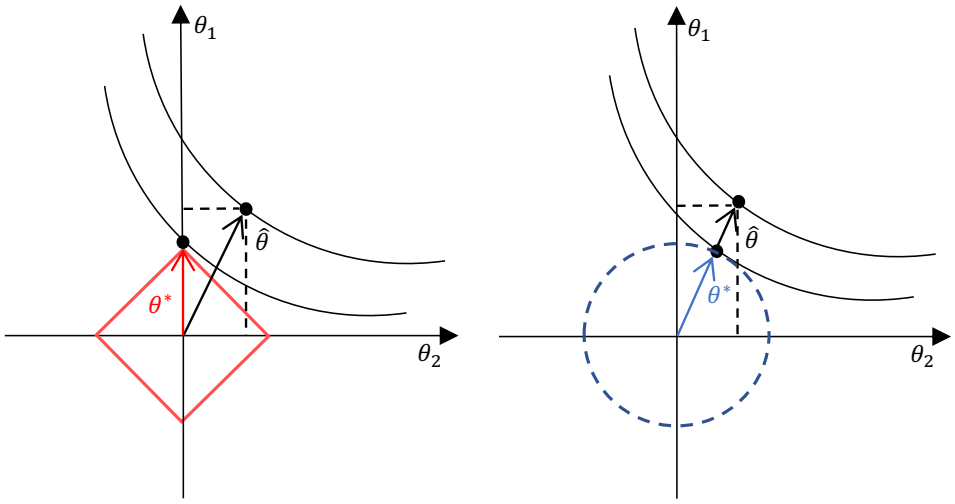
4.2.1 Regularization: Diminishing Utility Deterioration in High-dimensional Space

Regularization is a common technique to re-calibrate the minimization tasks [16, 18, 63,97,127]. On the one hand, it directly reduces the dimensions d. On the other hand, it also reduces the scale of the perturbed data and thus diminishes the variance, which counteracts the utility deterioration caused by high dimensionality [33].

To explain regularization, let $\mathcal{L}(\boldsymbol{\theta})$ denote a certain loss function regarding $\boldsymbol{\theta} \in \mathbb{R}^d$ while the regularization term is $\mathcal{R}(\boldsymbol{\theta})$. $\mathcal{R}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$ and $\mathcal{R} = \|\boldsymbol{\theta}\|_2$ are the operators for L_1-regularization (abbreviated as L_1) and L_2-regularization (abbreviated as L_2), respectively. Figure 4.2 illustrates the physical meaning of both regularizations in two dimensional space, where the black curves are isopleths of any loss function $\mathcal{L}(\boldsymbol{\theta})$. The red square is the shape of L_1, while the blue circle is the shape of L_2. We notice that $\mathcal{L}(\boldsymbol{\theta})$ converges to $\hat{\boldsymbol{\theta}}$ without regularization. In contrast to $\hat{\boldsymbol{\theta}}$, $\mathcal{L}(\boldsymbol{\theta})$ tends to cross on coordinate axes with L_1 while it tends to cross on the circle with L_2. Let $\boldsymbol{\theta}^*$ denote the regularized results. Comparing both $\boldsymbol{\theta}^*$ with $\hat{\boldsymbol{\theta}}$, L_1 reduces both dimensions and the scale of $\hat{\boldsymbol{\theta}}$ while L_2 just reduces the scale of $\hat{\boldsymbol{\theta}}$. By integrating them in the aggregation phase as a re-calibration, we can mitigate the negative influence by high dimensionality. In the next subsection, we propose our re-calibration protocol $HDR4ME*$.

4.2.2 HDR4ME*—High Dimensional Re-calibration for Mean Estimation

In general, we can take the estimation in high-dimensional space as an *empirical error minimization task*. We define the loss of function as $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2r}\sum_{i=1}^{r}\|\boldsymbol{t}_i^* - \boldsymbol{\theta}\|_2^2$. As such, the minimization task can be defined as $\hat{\boldsymbol{\theta}} = \arg\min_{\theta\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta})$. To minimize the error, we just obtain its derivative $\nabla\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{r}\sum_{i=1}^{r}\left(\boldsymbol{\theta} - \boldsymbol{t_i^*}\right) = \boldsymbol{\theta} - \frac{1}{r}\sum_{i=1}^{r}\boldsymbol{t}_i^*$ and



(a) L_1-regularization (b) L_2-regularization

Figure 4.2: Regularization in two dimensions.

make the derivative zero. Accordingly, we have $\hat{\boldsymbol{\theta}} = \frac{1}{r}\sum_{i=1}^{r} \boldsymbol{t}_i^*$. Note that this result is equivalent to that of the existing aggregation. Due to extremely large $Var(\boldsymbol{t}_{ij}^*)$, there are probably some extreme values in entries of $\boldsymbol{t}_i^*$, which greatly influence the direction of gradient descent and lead $\hat{\boldsymbol{\theta}}$ to deviate from the true mean $\bar{\boldsymbol{\theta}}$. In this context, we perform re-calibration on $\hat{\boldsymbol{\theta}}$ so as to reduce the gap between $\hat{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\theta}}$.

Recall that in each dimension, the data collector receives r perturbed tuples $\{\boldsymbol{t}_i^* | 1 \leq i \leq r\}$, where $r = \frac{nm}{d}$. To add regularization terms, we first define the loss function of the aggregation $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2r}\sum_{i=1}^{r} \|\boldsymbol{t}_i^* - \boldsymbol{\theta}\|_2^2$. On this basis, we add regularization terms $\mathcal{R}(\boldsymbol{\theta})$ to $\mathcal{L}(\boldsymbol{\theta})$ to obtain the enhanced mean $\boldsymbol{\theta^*}$ as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\theta \in \mathbb{R}^d} \{\mathcal{L}(\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\lambda}^* \circ \boldsymbol{\theta})\}, \tag{4.41}$$

where $\mathcal{R}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$ or $\|\boldsymbol{\theta}\|_2$ and $\boldsymbol{\lambda}^* = (\boldsymbol{\lambda}_1^*, ..., \boldsymbol{\lambda}_d^*)^\intercal$ is the regularization weight (which controls the degree of the involvement of regularization). In particular, $\boldsymbol{\lambda}^* \circ \boldsymbol{\theta} = (\boldsymbol{\lambda}_1^*\boldsymbol{\theta}_1, ..., \boldsymbol{\lambda}_d^*\boldsymbol{\theta}_d)^\intercal$ is *Hadamard product*. In what follows, we provide detailed utility analysis of $HDR4ME*$ with L_1- and L_2-regularization, respectively, together with the specification of $\boldsymbol{\lambda}^*$.

HDR4ME* with L_1-regularization. With this re-calibration, the deviation $\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2$ can be significantly reduced by dimensionality and perturbation reduction. The following lemma discusses the suitable choice of $\boldsymbol{\lambda}^*$ and the threshold for utility enhancement.

For L_1-regularization, it is pretty challenging for the regular gradient descent to obtain the enhanced mean $\boldsymbol{\theta}^*$ because $\mathcal{R} = \|\boldsymbol{\theta}\|_1$ is non-differentiable. As such, we adopt an alternative solution, namely, proximal gradient descent (PGD) [16, 79, 99]. In what follows, we introduce how PGD works in high-dimensional space. As a necessary prerequisite, we have to prove that $\nabla\mathcal{L}(\boldsymbol{\theta})$ satisfies *Lipschitz continuity* [131].

Lemma 4. HDR4ME* *with L_1-regularization can improve accuracy in j-th dimension if*

$$\boldsymbol{\lambda}_j^* = \sup\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| \ \ and \ \ \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 1 \tag{4.42}$$

where $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ is obtained from Lemma 2 or Lemma 3.

Proof. Since $\|\boldsymbol{\theta}\|_1$ is non-differentiable, we adopt an alternative solution, namely, proximal gradient descent (PGD) [16, 79, 99]. Our objective is to obtain the iterative equation to solve our protocol. First, we get the derivative of $\mathcal{L}(\boldsymbol{\theta})$:

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{r}\sum_{i=1}^{r}(\boldsymbol{\theta} - \boldsymbol{t}_i^*) = \boldsymbol{\theta} - \frac{1}{r}\sum_{i=1}^{r}\boldsymbol{t}_i^* = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \tag{4.43}$$

Thus, the derivative of $\nabla\mathcal{L}(\boldsymbol{\theta})$ is $\frac{\mathrm{d}\nabla\mathcal{L}(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}} = 1$. According to *Cauchy mean value theorem*, we have:

$$\|\nabla\mathcal{L}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta}_k)\|_2^2 \leq \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|_2^2, \tag{4.44}$$

where $\boldsymbol{\theta}_k$ is the result of k-th iteration. By *second-order Taylor expansion* around $\boldsymbol{\theta}_k$, we get:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\cong \mathcal{L}(\boldsymbol{\theta}_k) + \langle\nabla\mathcal{L}(\boldsymbol{\theta}_k), \boldsymbol{\theta} - \boldsymbol{\theta}_k\rangle + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|^2 \\ &= \frac{1}{2}\left\|\boldsymbol{\theta} - (\boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k))\right\|_2^2 + constant\end{aligned} \tag{4.45}$$

To minimize the loss function $\mathcal{L}$, we get the iterative equation $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k)$. We then introduce L_1-regularization term into the iteration:

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{\theta} - (\boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k))\|_2^2 + \|\boldsymbol{\lambda}^* \circ \boldsymbol{\theta}\|_1 \tag{4.46}$$

Since each dimension is independent of each other, we have the following solution for each dimension:

$$(\boldsymbol{\theta}_{k+1})_j = \arg\min_{\boldsymbol{\theta}_j} \frac{1}{2} |\boldsymbol{\theta}_j - ((\boldsymbol{\theta}_k)_j - \nabla\mathcal{L}(\boldsymbol{\theta}_k)_j)|_2^2 + \left|\boldsymbol{\lambda}_j^* \boldsymbol{\theta}_j\right|_1 \tag{4.47}$$

As such, how to compute $(\boldsymbol{\theta}_{k+1})_j$ really depends on whether $\boldsymbol{\theta}_j$ is positive, zero or negative ($\boldsymbol{\lambda}_j$ is positive). In particular, we let $\boldsymbol{z} = \boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k) = \boldsymbol{\theta}_k - \boldsymbol{\theta}_k + \hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}$. If $\boldsymbol{\theta}_j > 0$, we get the gradient of Equation 4.47 as $\boldsymbol{\theta}_j - \boldsymbol{z}_j + \boldsymbol{\lambda}_j^*$. By making it zero, we obtain $(\boldsymbol{\theta}_{k+1})_j = \boldsymbol{z}_j - \boldsymbol{\lambda}_j^* > 0$, in which case $\boldsymbol{z}_j > \boldsymbol{\lambda}_j^*$. If $\boldsymbol{\theta}_j < 0$, we similarly obtain $(\boldsymbol{\theta}_{k+1})_j = \boldsymbol{z}_j + \boldsymbol{\lambda}_j^* < 0$, in which case $\boldsymbol{z}_j < -\boldsymbol{\lambda}_j^*$. If $\boldsymbol{\theta}_j = 0$, Equation 4.47 simply converges and $(\boldsymbol{\theta}_{k+1})_j = 0$, which corresponds with $|\boldsymbol{z}_j| \leq \boldsymbol{\lambda}_j^*$. Accordingly, we have the following iteration:

$$(\boldsymbol{\theta}_{k+1})_j = \begin{cases} \boldsymbol{z}_j - \boldsymbol{\lambda}_j^*, & \boldsymbol{z}_j > \boldsymbol{\lambda}_j^* \\ 0, & |\boldsymbol{z}_j| \leq \boldsymbol{\lambda}_j^* \\ \boldsymbol{z}_j + \boldsymbol{\lambda}_j^*, & \boldsymbol{z}_j < -\boldsymbol{\lambda}_j^* \end{cases} \tag{4.48}$$

Since $\boldsymbol{z}$ and $\boldsymbol{\lambda}^*$ are deterministic, Equation 4.48 is actually a one-off solver. If we set $\boldsymbol{\lambda}_j^* = \sup\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$, for $\boldsymbol{\theta}_j^* > 0$, we have $\boldsymbol{\theta}_j^* = \boldsymbol{z}_j - \boldsymbol{\lambda}_j^* = \hat{\boldsymbol{\theta}}_j - \sup\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| \leq \hat{\boldsymbol{\theta}}_j - \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$. Since $\boldsymbol{\theta}_j^*$ is re-calibrated from $\boldsymbol{z}_j = \hat{\boldsymbol{\theta}}_j$, $\hat{\boldsymbol{\theta}}_j$ has the same sign as $\boldsymbol{\theta}_j^*$, which implies $\hat{\boldsymbol{\theta}}_j > \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$. Suppose $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 1$, which happens frequently in high-dimensional space, we then have $\hat{\boldsymbol{\theta}}_j > 1$. Because $\bar{\boldsymbol{\theta}}_j \leq 1$, $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| = \hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ holds. Therefore, we have $0 < \boldsymbol{\theta}_j^* \leq \hat{\boldsymbol{\theta}}_j - \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| = \bar{\boldsymbol{\theta}}_j \leq 1$, which proves $0 \leq \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < 1$. As such, $\left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < 1 < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$ accordingly holds. Similarly, we derive $\left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < 1 < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$ for $\boldsymbol{\theta}_j^* < 0$. For $\boldsymbol{\theta}_j^* = 0$, $\left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| = \left|\bar{\boldsymbol{\theta}}_j\right| \leq 1 < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$. In general,

we have:

$$\begin{aligned} \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| &< \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| \\ \text{if } \boldsymbol{\lambda}_j^* = \sup \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| &\text{ and } \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 1 \end{aligned} \tag{4.49}$$

□

This lemma specifies the suitable regularization weight and the required threshold for utility enhancement in one dimension. On this basis, we prove the superiority of HDR4ME* with L_1 to the current aggregation in high-dimensional space.

Theorem 4. *For any high-dimensional LDP mechanism $\mathcal{M}$ under* HDR4ME* *with L_1-regularization, the following inequality holds with at least $1 - \int_{-1}^{1} ... \int_{-1}^{1} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\mathrm{d}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$ probability:*

$$\|\boldsymbol{\theta}^* - \bar{\boldsymbol{\theta}}\|_2 < \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \tag{4.50}$$

where $f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$ is obtained from Theorem 1.

Proof. Lemma 4 establishes that $\left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$ holds with one certain threshold: $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 1$. Theorem 1 derives that $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 1$ holds for $\forall j \in [1, d]$ with at least the probability $1 - \int_{-1}^{1} ... \int_{-1}^{1} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\mathrm{d}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$. On this basis,

$$\|\boldsymbol{\theta}^* - \bar{\boldsymbol{\theta}}\|_2 = \sqrt{\sum_{j=1}^{d} \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right|^2} < \sqrt{\sum_{j=1}^{d} \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|^2} = \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \tag{4.51}$$

□

In general, Theorem 4 derives the least probability for L_1 to enhance utilities in high-dimensional space. Nevertheless, a solver to HDR4ME* with L_1 is still required. Applying $\boldsymbol{z} = \hat{\boldsymbol{\theta}}$ and $(\boldsymbol{\theta}_{k+1})_j = \boldsymbol{\theta}_j^*$ to Equation 4.48, we have:

$$\boldsymbol{\theta}_j^* = \begin{cases} \hat{\boldsymbol{\theta}}_j - \boldsymbol{\lambda}_j^*, & \hat{\boldsymbol{\theta}}_j > \boldsymbol{\lambda}_j^* \\ 0, & \left|\hat{\boldsymbol{\theta}}_j\right| \leq \boldsymbol{\lambda}_j^* \\ \hat{\boldsymbol{\theta}}_j + \boldsymbol{\lambda}_j^*, & \hat{\boldsymbol{\theta}}_j < -\boldsymbol{\lambda}_j^* \end{cases} \tag{4.52}$$

Equation 4.52 is a one-off, non-iterative solver for HDR4ME* with L_1, which simply re-calibrates the estimated mean to get the enhanced mean. As such, the data collector can enhance utilities without bearing extra computational burden.

HDR4ME* with L_2-regularization. This re-calibration can obtain much deviation $\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2$ by scale reduction. To achieve this, $\boldsymbol{\lambda}^*$ must satisfy the following condition.

Lemma 5. HDR4ME* *with L_2-regularization can improve accuracy in j-th dimension if*

$$\boldsymbol{\lambda}_j^* = \sup \frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j}{2\bar{\boldsymbol{\theta}}_j} \text{ and } \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 2 \tag{4.53}$$

where $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ is obtained from Lemma 2 or Lemma 3, and $\bar{\boldsymbol{\theta}}_j$ can select the mean of the normal distribution that approximates $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ in our framework.

Proof. Following Equation 4.45, we add L_2-regularization term into the iteration:

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{\theta} - (\boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k))\|_2^2 + \|\boldsymbol{\lambda}^* \circ \boldsymbol{\theta}\|_2^2 \tag{4.54}$$

Since each dimension is perturbed independently, we have the following solution for each dimension:

$$(\boldsymbol{\theta}_{k+1})_j = \arg\min_{\boldsymbol{\theta}_j} \frac{1}{2} |\boldsymbol{\theta}_j - ((\boldsymbol{\theta}_k)_j - \nabla\mathcal{L}(\boldsymbol{\theta}_k)_j)|_2^2 + \left|\boldsymbol{\lambda}_j^* \boldsymbol{\theta}_j\right|^2 \tag{4.55}$$

where $\boldsymbol{z} = \boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k) = \hat{\boldsymbol{\theta}}$. Note that Equation 4.55 is differentiable. Applying 0 to the derivative of Equation 4.55, we have $\boldsymbol{\theta}_j^* = \frac{\hat{\boldsymbol{\theta}}_j}{2\boldsymbol{\lambda}_j^* + 1}$. If $\boldsymbol{\lambda}_j^* = \sup \frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j}{2\bar{\boldsymbol{\theta}}_j}$, our framework implies $\boldsymbol{\lambda}_j^* > 0$. Then, we derive:

$$|\boldsymbol{\theta}_j^*| = \left|\frac{\hat{\boldsymbol{\theta}}_j}{2\boldsymbol{\lambda}_j^* + 1}\right| = \left|\frac{\hat{\boldsymbol{\theta}}_j}{\sup \frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j}{\bar{\boldsymbol{\theta}}_j} + 1}\right| \leq \left|\frac{\hat{\boldsymbol{\theta}}_j}{\frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j}{\bar{\boldsymbol{\theta}}_j} + 1}\right| = \left|\bar{\boldsymbol{\theta}}_j\right| \tag{4.56}$$

Therefore, we have $0 \leq |\boldsymbol{\theta}_j^*| \leq \left|\bar{\boldsymbol{\theta}}_j\right| \leq 1$, which implies $0 \leq \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| \leq 2$. Namely, we have:

$$\begin{gathered} \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| \\ \text{if } \boldsymbol{\lambda}_j^* = \sup \frac{\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j}{2\bar{\boldsymbol{\theta}}_j} \text{ and } \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 2 \end{gathered} \tag{4.57}$$

□

Now that this lemma specifies the suitable regularization weight and the required threshold for utility enhancement in one dimension, we further prove the superiority of HDR4ME* with L_2 to the current aggregation in high-dimensional space.

Theorem 5. *For any high-dimensional LDP mechanism $\mathcal{M}$ under* HDR4ME* *with L_2-regularization, the following inequality holds with at least $1 - \int_{-2}^{2} ... \int_{-2}^{2} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\mathrm{d}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$ probability:*

$$\|\boldsymbol{\theta}^* - \bar{\boldsymbol{\theta}}\|_2 < \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \tag{4.58}$$

where $f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$ is obtained from Theorem 1.

Proof. Equation 4.49 in Lemma 5 derives that $\left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right| < \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|$ holds with one certain threshold: $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 2$. Theorem 1 derives that $\left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right| > 2$ holds for $\forall j \in [1, d]$ with at least the probability $1 - \int_{-2}^{2} ... \int_{-2}^{2} f(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\mathrm{d}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$. On this basis,

$$\|\boldsymbol{\theta}^* - \bar{\boldsymbol{\theta}}\|_2 = \sqrt{\sum_{j=1}^{d} \left|\boldsymbol{\theta}_j^* - \bar{\boldsymbol{\theta}}_j\right|^2} < \sqrt{\sum_{j=1}^{d} \left|\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j\right|^2} = \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \tag{4.59}$$

□

With our framework, Theorem 5 derives the least probability for L_2 to enhance utilities in high-dimensional space, in which case the enhanced mean is always better than estimated mean. To solve HDR4ME* with L_2, we compute the derivative of Equation 4.55 and set it to zero:

$$\boldsymbol{\theta}_j^* = \boldsymbol{\theta}_k - \nabla\mathcal{L}(\boldsymbol{\theta}_k) = \frac{\hat{\boldsymbol{\theta}}_j}{2\boldsymbol{\lambda}_j^* + 1} \tag{4.60}$$

Similarly, the above is also a one-off, non-iterative solver for HDR4ME* with L_2, which does not increase the computational burden of the data collector.

As a final note, both types of *HDR4ME** are designed for "high-dimensional" space only. In such a space, the useful statistics are flooded by much larger noise, which

provides us room to make utility enhancement. If the number of dimensions is not high or the collective privacy budget is rather large, which generally means that the threshold for either regularization to enhance utilities is not reached, our re-calibration can be harmful.

Now that we provide the theoretical foundations of both regularizations. In terms of utility, according to [30], L_2 **enhances utility better than** L_1. With either regularization, the enhanced mean is meant to be smaller than the estimated mean because making either regularization term small enough is part of the minimization task. But how small it is differs. Under the same task, the absolute regularization term $\|\boldsymbol{\theta}\|_1$ is much smaller than the squared regularization term $\|\boldsymbol{\theta}\|_2$, thus incurring more zeros in dimensions. For example, under the same enhanced mean in one dimension $\boldsymbol{\theta}_j^* = 0.1$, the L_2-regularization term $\left|\boldsymbol{\theta}_j^*\right|^2 = 0.01$ while L_1-regularization term $\left|\boldsymbol{\theta}_j^*\right| = 0.1$. That is, L_1 tends to push the enhanced mean to an extremely small value, i.e. close to 0 whereas the enhanced mean in L_2 can be a none-zero value. This observation can also be validated by the solution to either L_1-regularization (Equation 4.52) or L_2-regularization (Equation 4.60). Note that as the enhanced mean by L_1-regularization reaches 0, it means that the regularization term overwhelms the true information embedded in the perturbed data. In contrast, L_2-regularization still preserves such information since its enhanced mean never approaches 0. Nevertheless, L_2 has its weakness. Lemma 5 shows that L_2 is effective only if the deviation is larger than 2, but this threshold is 1 for L_1 according to Lemma 4. As such, **the prerequisite for L_2-regularization to enhance utility is more stringent than L_1-regularization**. Weighing the pros and cons between L_1 and L_2 is worth further investigation.

In essence, both regularizations work when the true mean is not very large while the estimated mean is too large due to the excessive perturbation noise in high dimensional space. By contrast, if the dimensionality is not high or the collective privacy budget is very large, that is, the deviation threshold for neither regulariza-

tion to enhance utilities (1 or 2) is reached, no regularization should be used. To maximize the utility enhancement based on our former discussion, we would like to implement L_1-regularization only if the deviation is larger than 1 but no more than 2, L_2-regularization only if the deviation is larger than 2 and existing aggregation only if the deviation is no more than 1. Considering no solution always dominates in high-dimensional space, we propose *HDR4ME** in Algorithm 4 to adaptively decide in each dimension which strategy (L_1, L_2, or none-regularization) should be adopted. Supported by our framework, in each dimension, it first calculates the probabilities (p_1, p_2 and p_3) for L_1, L_2 and non-regularization to be effective in line 2, respectively. Denoted as $\boldsymbol{\theta}_j^{e*}$, $\boldsymbol{\theta}_j^{1*}$ and $\boldsymbol{\theta}_j^{2*}$, Line 3 next computes the enhanced means for L_1 (from Equation 4.52), L_2 (from Equation 4.60) and non-regularization (from the existing aggregation), respectively. As presented in line 4, the weighted average of three enhanced means is considered the optimal.

Algorithm 4 HDR4ME*

Input: the estimated mean $\hat{\boldsymbol{\theta}}$, both regularization weights $\boldsymbol{\lambda}^*$.

Output: the enhanced mean $\boldsymbol{\theta}^*$.

1: **for** $j = 1$ to d **do**

2: generate weights for non-regularization:

$$p_1 = \int_{-1}^{1} f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) d(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j)$$

L_1-regularization:

$$p_2 = \int_{-2}^{2} f(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) d(\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j) - p_1$$

L_2-regularization:

$$p_3 = 1 - p_1 - p_2$$

3: derive the enhanced means for non-regularization: $\boldsymbol{\theta}_j^{e*}$, L_1-regularization: $\boldsymbol{\theta}_j^{1*}$, L_2-regularization: $\boldsymbol{\theta}_j^{2*}$

4: compute $\boldsymbol{\theta}_j^* = p_1\boldsymbol{\theta}_j^{e*} + p_2\boldsymbol{\theta}_j^{1*} + p_3\boldsymbol{\theta}_j^{2*}$

5: **end forreturn** $\boldsymbol{\theta}^*$

4.2.3 High-dimensional Re-calibration for Frequency Estimation

For various LDP mechanisms, high-dimensional frequency estimation is never sufficiently discussed, especially when some mechanisms claim to be applicable to both mean and frequency estimations [134, 137]. As such, we also generalize our re-calibration to frequency estimation. Note that any categorical value can be mapped into a binary vector with *histogram encoding* [137]. Suppose there are d categorical dimensions and $v_j (1 \leq j \leq d)$ categories in each dimension, any categorical value in j-th dimension $\boldsymbol{t}_{ij} (1 \leq i \leq r_j)$ can be encoded to a v_j-entry vector $(0.0, 0.0, ..., 1.0, ..., 0.0)^\intercal$ with only the $\boldsymbol{t}_{ij}$-th entry to be 1.0. As such, each of d categorical dimensions is expanded to one v_j-dimensional numerical space. Note that each entry of encoded vectors ranges from $[0, 1]$. If each user reports m dimensions of her perturbed data to the data collector, the collective ϵ-LDP can be guaranteed by applying $\frac{\epsilon}{2m}$ to each entry of vectors [137] regardless of LDP mechanisms. As such, the data collector receives r_j v_j-entry perturbed vectors in j-th dimension. Since each entry corresponds with one certain categorical value, the mean of r_j perturbed vectors corresponds with the estimated frequencies in j-th dimension, with each entry of the mean to be the frequency of each categorical value. In general, we can convert one d-dimensional frequency estimation to d high-dimensional mean estimation tasks. On this basis, both our framework and re-calibration protocol can further apply.

4.3 Empirical Results

To verify both the analytical framework and the re-calibration protocol, we conduct experiments under a real dataset COV-19[2] and three synthetically distributed datasets, namely Gaussian, Poisson and Uniform. The following are some descriptions

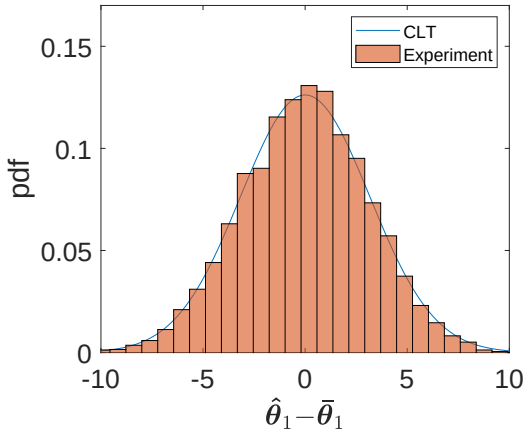
[2]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

of four datasets:

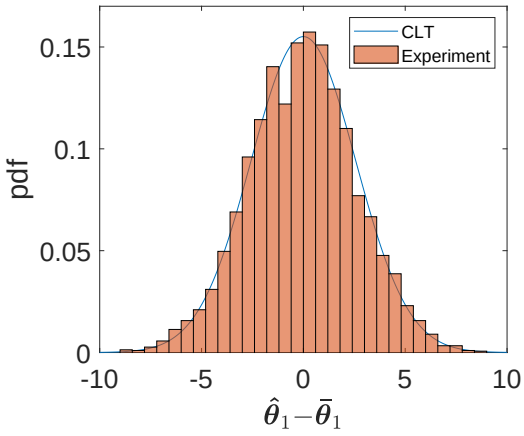
- The COV-19 dataset consists of 150,000 users and 750 dimensions, where each dimension has high correlations with others.
- The Gaussian dataset consists of tunable users and dimensions. The standard deviation of all dimensions is set to 1/16. 10% dimensions have their mathematical expectations $\mu = 0.9$ whereas the other 90% have $\mu = 0$.
- The Poisson dataset consists of 150,000 users and 300 dimensions, where each dimension follows a Poisson distribution with a random expectation from 1 to 99.
- The Uniform dataset consists of tunable users and dimensions.

The aims of our experiments are twofold. First, we confirm the effectiveness of our analytical framework, namely, $\hat{\boldsymbol{\theta}}_j - \bar{\boldsymbol{\theta}}_j$ can be approximated with one certain Gaussian distribution. Second, we compare the performances of *HDR4ME* on top of the aggregation results of three state-of-the-art LDP mechanisms, i.e., Laplace [35], Piecewise [134], and Square wave [83]. Each dimension is normalized into $[-1, 1]$, and each experiment is repeated 100 times to obtain the averaged result unless otherwise indicated. All our experiments are implemented in MATLAB on a laptop computer with Intel Core i7-10750H 2.59 GHz CPU, 32G RAM on Windows 10 operation system.

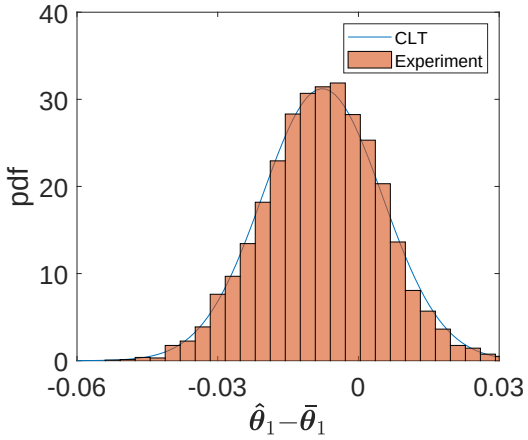
To start with, in the first set of experiments, we use Uniform dataset to verify the effectiveness of our analytical framework in terms of regular LDP, where we set 200,000 users and 5,000 dimensions. Each user sends 50 dimensions of her perturbed tuples to the data collector. Each experiment is iterated 1,000 times, and we collect the means of 1,000 times in the first dimension. Given the collective privacy budget $\epsilon = 1$, Fig. 4.3 shows how our framework models the means from experiments. In each sub-figure, the blue line is the pdf of the deviations from our framework while the orange squares are the pdf estimate from experiments. In all three mechanisms,



(a) Laplace



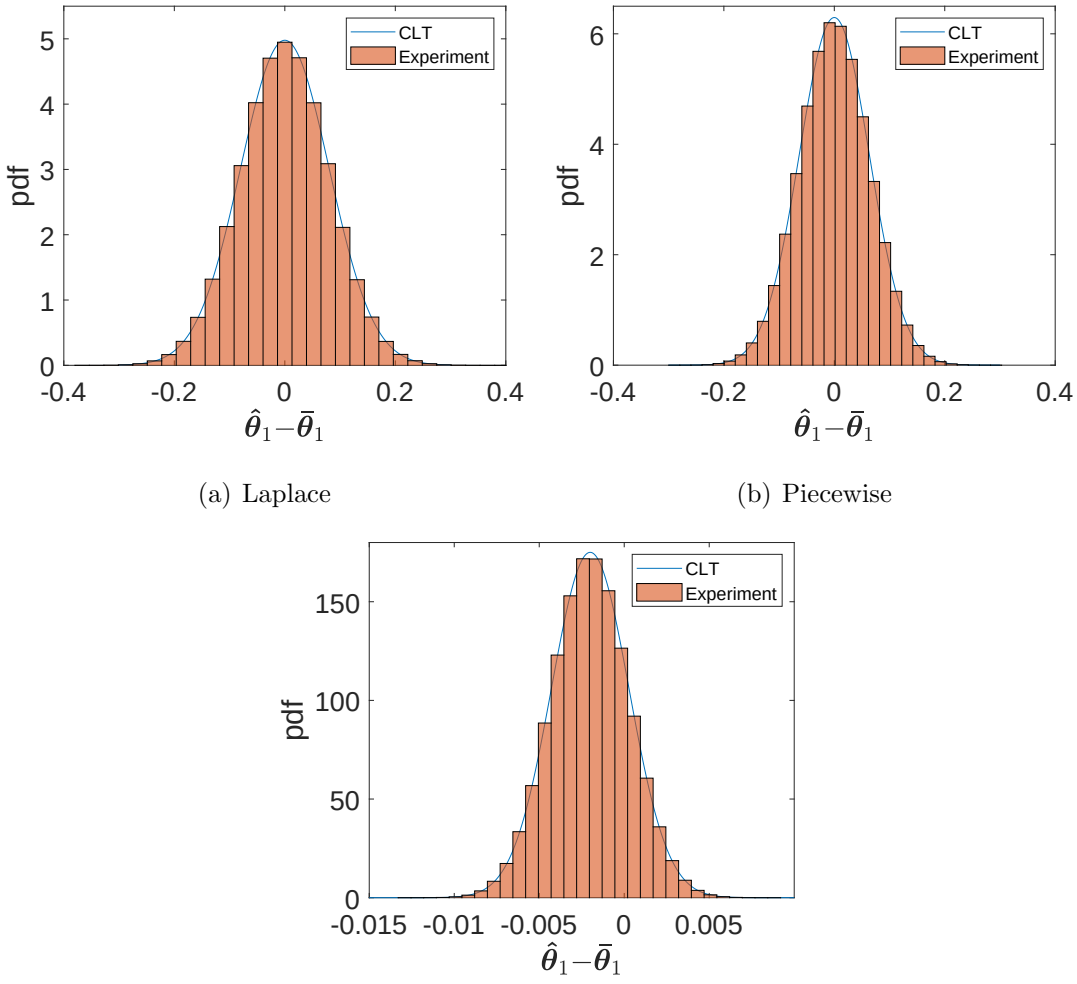
(b) Piecewise



(c) Square

Figure 4.3: Analysis vs. experimental results on Uniform dataset under regular LDP (d=5,000).

our framework effectively approximates experimental results. Recall that we provide a case study in Section 6.1.4 to benchmark Piecewise and Square wave. To support the benchmark results, we discretize the Uniform dataset and plot in Fig. 4.5. In both mechanisms, the pdf functions computed in our case study perfectly align with the experimental results, which confirms the effectiveness of the benchmark by our framework.



(a) Laplace

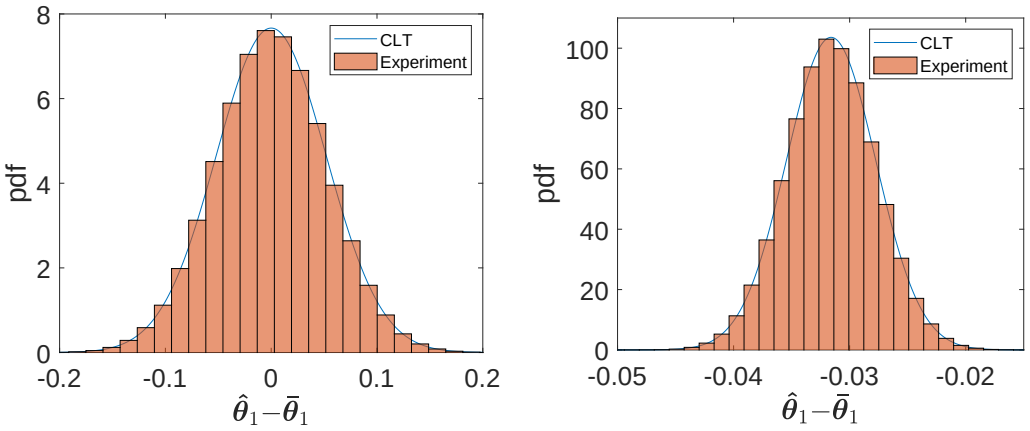
(b) Piecewise

(c) Square

Figure 4.4: Analysis vs. experimental results on Uniform dataset under personalized LDP (d=2,000).

Similarly, we use Uniform dataset to evaluate the effectiveness of our framework

in PLDP. In detail, number of users is set 1,000,000 and that of dimensions is set 2,000. For each user, the available privacy budgets and privacy regions are $\{\epsilon_1 = 1, \epsilon_2 = 5, \epsilon_3 = 10, \epsilon_4 = 15, \epsilon_5 = 20\}$ and $\tau_1 = [-1.0, 0), \tau_2 = [0, 1.0]$, respectively. As such, there are $5 \times 2 = 10$ subgroups. Without loss of generality, each user uniformly chooses one budget and one region, so we set 1,000,000/10=100,000 users in each subgroup. As each user sends 20 dimensions of her perturbed tuple to the data collector in each iteration, we iterate experiments 1,000 times on Laplace, Piecewise and Square wave, respectively. Finally, we collect the means of 1,000 times in the first dimension and therefore illustrate how our framework models experimental means for PLDP in Fig. 4.4. In each sub-figure, the theoretical distribution (the blue line) effectively matches with experimental results (the orange squares), which confirms the generality of our framework in PLDP.



(a) Piecewise (b) Square

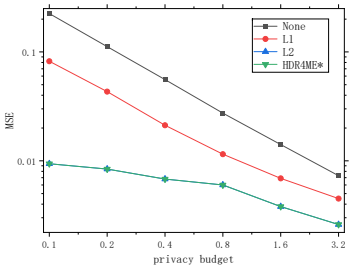
Figure 4.5: Analysis vs. experimental results in our case study.

In the second set of experiments, we compare HDR4ME* with HDR4ME and the existing aggregation under different LDP mechanisms, privacy budget and dimensionality. In particular, ϵ is varied in the set $\{0.1, 0.2, 0.4, 0.8, 1.6, 3.2\}$ for Laplace and Piecewise while in the set $\{0.1, 10, 100, 500, 1000, 5000\}$ for Square wave. We set a different range of ϵ for Square wave because its utility hardly varies with small ϵ [83].

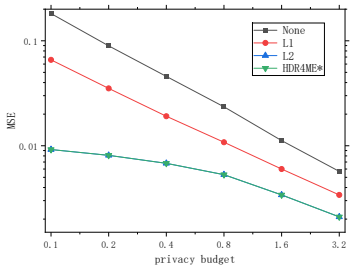
To test the limit of our candidates, each user sends all dimensions of her perturbed tuple to the data collector. Accordingly, ϵ is partitioned according to respective dimensions. While using black, red, blue and green lines to reflect performances of Non-, L_1-, L_2-regularization and HDR4ME*, respectively, Figs. 4.6(a)-(c) plot MSE results with respect to ϵ under the Gaussian dataset, where users and dimensions are respectively set 100,000 and 100. Overall, HDR4ME* performs no worse than L_1-, L_2 or none-regularization. In face of Laplace and Piecewise mechanisms which have low utilities in high-dimension, HDR4ME* tends to select the best among L_1 and L_2 rather than non-regularization, as indicated in Fig. 4.6. To avoid harm to LDP mechanisms with possibly small deviations, such as Square wave, HDR4ME* however puts heavy weight to non-regularization while still considering the necessity of regularizations.

Similarly, we implement Poisson dataset (150,000 users, 300 dimensions), Uniform dataset (120,000 users, 500 dimensions) and COV-19 dataset (150,000 users, 750 dimensions) to repeat the above experiments. ϵ is also partitioned according to respective dimensions. Figs. 4.6(d)-(f), (g)-(i) and (j)-(l) show respective MSE results with regard to different datasets. On contrast, Figs. 4.6(d), (e), (g), (h), (j) and (k) still confirm that HDR4ME* can ensure the optimal utility if both regularizations are beneficial while Figs. 4.6(f), (i) and (l) reveal that HDR4ME* can still adaptively optimize utility even both regularizations fail. Similar results from various real datasets can be observed in Fig. 4.8.

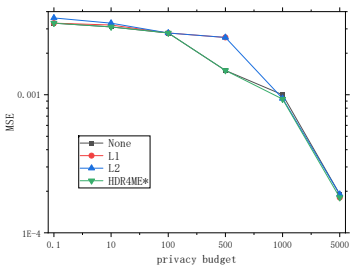
In the third sets of experiments, we evaluate HDR4ME* and HDR4ME under the COV-19 dataset, where ϵ is set 0.8, and the dimensionality varies in $\{50, 100, 200, 400, 800, 1600\}$. Since the dataset with dimensionality like 1600 is very hard to find, we randomly sample some dimensions from COV-19 dataset to make up. Fig. 4.7 shows MSE results of the Laplace and Piecewise, where HDR4ME* adaptively favors L_2-regularization as it dominates the enhancement in face of LDP mechanisms with large deviations, regardless of dimensionality. As L_2 excels, $HDR4ME*$ also achieves much better



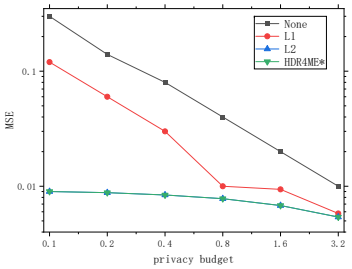
(a) Laplace on Gaussian ($d = 100$)



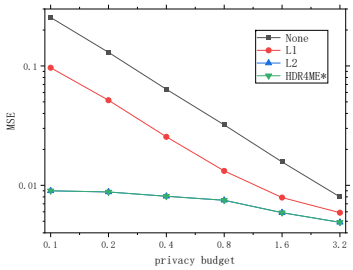
(b) Piecewise on Gaussian ($d = 100$)



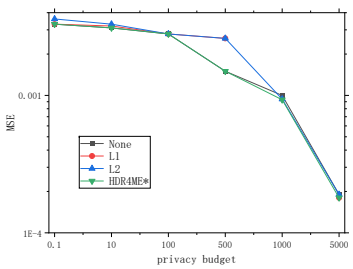
(c) Square on Gaussian ($d = 100$)



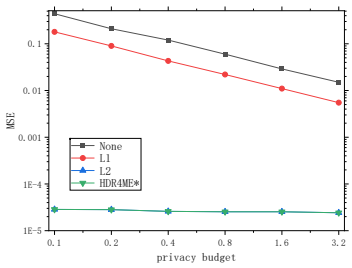
(d) Laplace on Poisson ($d = 300$)



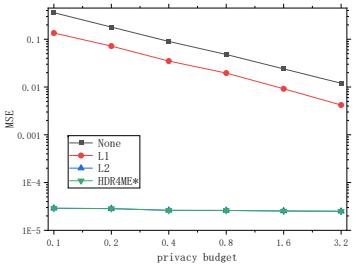
(e) Piecewise on Poisson ($d = 300$)



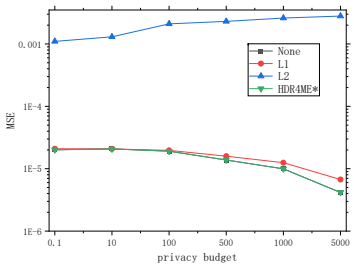
(f) Square on Poisson ($d = 300$)



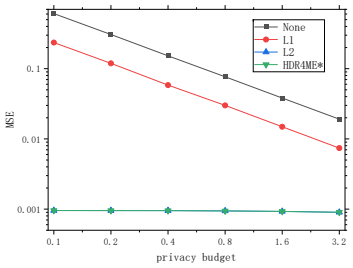
(g) Laplace on Uniform ($d = 500$)



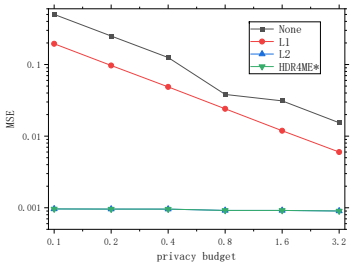
(h) Piecewise on Uniform ($d = 500$)



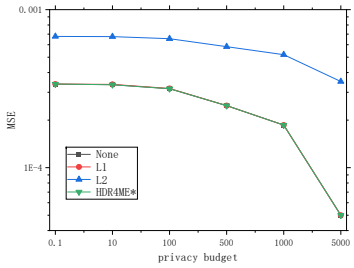
(i) Square on Uniform ($d = 500$)



(j) Laplace on COV-19 ($d = 750$)



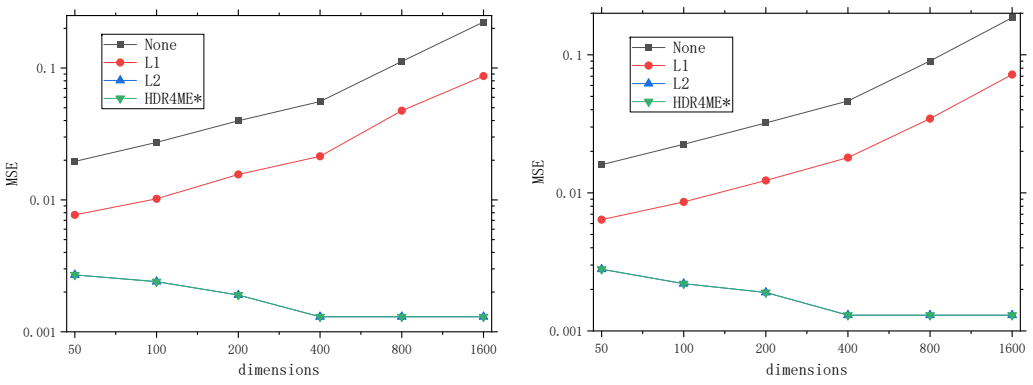
(k) Piecewise on COV-19 ($d = 750$)



(l) Square on COV-19 ($d = 750$)

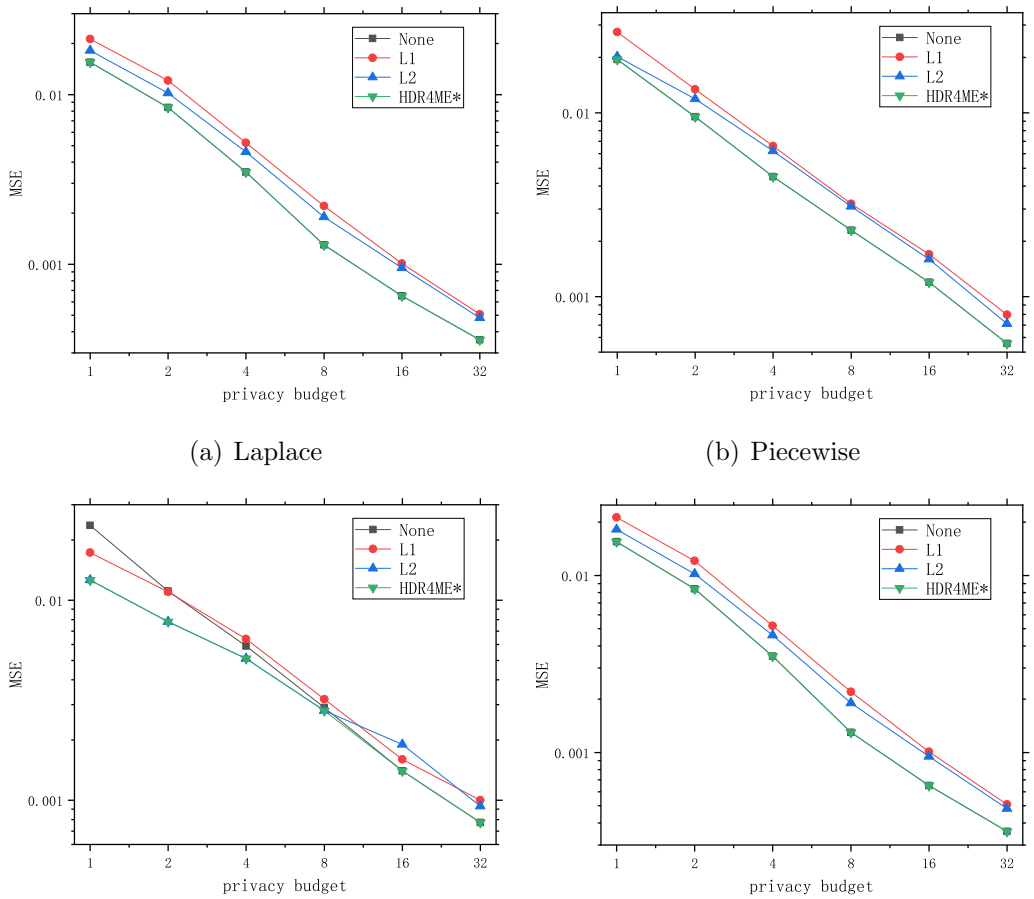
Figure 4.6: MSE on various datasets and dimensions

utilities, as opposed to both the current aggregation and L_1. Most likely, the regularization weights of L_2 are much larger than those of L_1 as dimensionality increases, which reduces the scale of perturbation more effectively. In this sense, MSE results of L_2 in both mechanisms decrease as dimensionality increases (e.g. $d = 50, 100, 200$). As the dimensionality becomes extremely large (e.g. $d = 400, 800, 1600$), regularization weights of L_2 become so large that each entry of enhanced mean is nearly zero. In this sense, MSE results of L_2 hardly change, so is it for HDR4ME*.



(a) Laplace (b) Piecewise

Figure 4.7: MSE on COV-19 dataset with various dimensions.



(a) Laplace (b) Piecewise

(c) Laplace (d) Piecewise

Figure 4.8: MSE on Real Datasets.

4.4 Summary

This chapter investigates utilities of mean estimation by LDP mechanisms in high-dimensional space. In terms of the deviation between the estimated mean and the true mean, we propose an analytical framework to evaluate any LDP mechanism. This framework provides closed-form evaluation on individual LDP mechanism. In addition, we propose *HDR4ME* to re-calibrate the aggregation results from these LDP mechanisms to further enhance their utilities in high-dimensional space. Through

theoretical analysis and extensive experiments, we confirm the generality and effectiveness of our analytical framework and re-calibration protocol under various datasets and parameter settings.

For the future work, we plan to extend our work to other data type, e.g., set-value data, and more data analysis tasks, e.g., other statistics estimation and machine learning models. In the future, we would like to extend our work to multi-dimensional and one-dimensional space.

Chapter 5

Analyzing and Optimizing Perturbation of DP-SGD Geometrically

Although deep learning models have numerous applications in various domains, such as personal recommendation and healthcare, the privacy leakage of training data from these models has become a growing concern. There are already mature attacks which successfully reveal the contents of private data from deep learning models [20, 54]. For example, a white-box membership inference attack can infer whether a single data point belongs to the training dataset of a DenseNet with 82% test accuracy [94]. These attacks pose imminent threats to the wider adoption of deep learning in business sectors with sensitive data, such as healthcare and fintech.

To address this concern, differential privacy (DP), which can provide quantitative amount of privacy preservation to individuals in the training dataset, is embraced by the most prevalent optimization technique of model training, i.e., stochastic gradient descent (SGD). Referred to as DP-SGD [6, 82, 85, 160], this algorithm adds random DP noise to gradients in the training process so that attackers cannot infer private

data from model parameters with a high probability.

However, a primary drawback of DP-SGD is the ineffective training process caused by the overwhelming noise, which extremely deteriorates the model efficiency. Although much attention [1, 46, 92] has been paid on reducing the noise scale, the majority of existing solutions, which numerically add DP noise to gradients, do not exploit the geometric nature of SGD (i.e., descending gradient to locate the optima). As reviewed in Section 2.2.3, SGD exhibits a distinctive geometric property — the direction of a gradient rather than the magnitude determines the descent trend. By contrast, regular DP algorithms, such as the Gaussian mechanism [36], was originally designed to preserve numerical (scalar) values rather than vector values. As such, there is a distinct gap between directional SGD and numerical DP perturbation, causing at least two limitations in DP-SGD. First, **existing optimization techniques of SGD (i.e., fine-tuning clipping and learning rate)**, which can effectively reduce the noise on the magnitude of a gradient, **cannot alleviate the negative impact on the direction**, as illustrated by Example 1. Second, **traditional DP introduces biased noise on the direction of a gradient**, even if the total noise to the gradient is unbiased (proved in Lemma 6). As a result, the perturbation of traditional DP-SGD is only sub-optimal from a geometric perspective.

Example 1. *Suppose that we have a two-dimensional gradient* $\boldsymbol{g} = (1, \sqrt{3})$ *with its direction* $\boldsymbol{\theta} = \arctan(\sqrt{3}/1) = \pi/3$ *and magnitude* $\|\boldsymbol{g}\| = \sqrt{1+3} = 2$. *Given clipping threshold* $C_1 = 2$, *we add noise* $\boldsymbol{n}_1 = (0.3, 0.15)$ *to the clipped gradient* $\tilde{\boldsymbol{g}}_1 = \boldsymbol{g}/\max\{1, \|\boldsymbol{g}\|/C_1\} = (1, \sqrt{3})$ *and derive the perturbed direction* $\boldsymbol{\theta}_1^* = \arctan\frac{\sqrt{3}+0.15}{1+0.3} \approx 0.97$. *If* $C_2 = 1$, *the clipped gradient and the noise would be* $\tilde{\boldsymbol{g}}_2 = \boldsymbol{g}/\max\{1, \|\boldsymbol{g}\|/C_2\} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ *and* $\boldsymbol{n}_2 = \boldsymbol{n}_1/(C_1/C_2) = (0.15, 0.075)$, *respectively, as per DP-SGD [1]. Still, the perturbed direction is* $\boldsymbol{\theta}_2^* = \arctan\frac{\frac{\sqrt{3}}{2}+0.075}{\frac{1}{2}+0.15} \approx 0.97$. *Although the noise scale is successfully reduced by gradient clipping (*$\|\boldsymbol{n}_2\| < \|\boldsymbol{n}_1\|$*), the perturbation on the direction of a gradient remains the same (*$\boldsymbol{\theta}_2^* = \boldsymbol{\theta}_1^*$*).*

In this chapter, we propose a geometric perturbation strategy GeoDP to address these

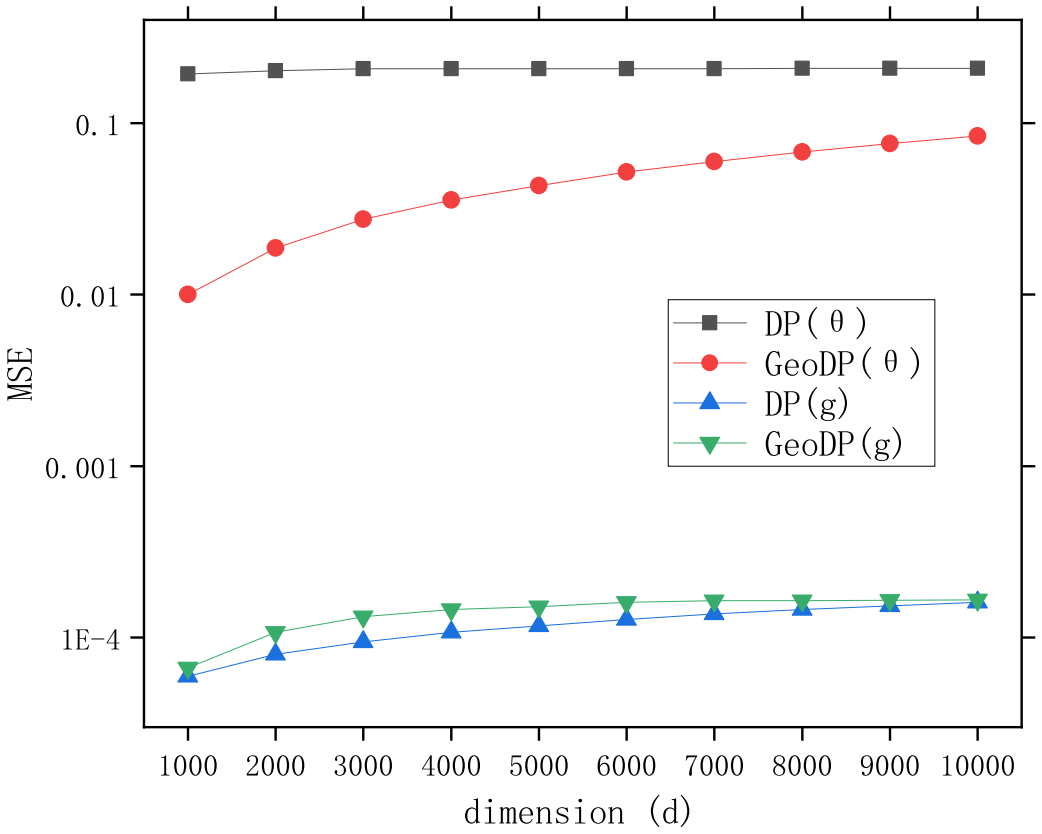


Figure 5.1: Comparing MSEs of GeoDP and DP on preserving directions and values of gradients under synthetic dataset (composed of gradients from CNN training, as introduced in Section 5.3.1). While θ and g label the MSE of perturbed directions and gradients themselves, experimental results confirm that GeoDP achieves smaller MSEs on perturbed directions (i.e., the red line is below the black one), while sacrificing the accuracy of perturbed gradients (i.e., the green lien is above the blue one). In general, GeoDP better preserves directions of gradients while traditional DP only excels in preserving numerical values of gradients.

limitations. First, we theoretically derive the impact of DP noise on the efficiency of DP-SGD. Proved by this fine-grained analysis, the perturbation of DP-SGD, which introduces biased noise to the direction of a gradient, is actually sub-optimal. Inspired by this, we propose a geometric perturbation strategy *GeoDP* which perturbs both the direction and the magnitude of a gradient, so as to relieve the noisy gradient direction and optimize model efficiency with the same DP guarantee. Figure 5.1 illustrates empirical performances of GeoDP and DP to support the superiority of GeoDP in the perspective of geometry. Such experimental results can also be confirmed in our theoretical analysis. Frequently-used notations are listed in Table 5.1. In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to prove that the perturbation of traditional DP-SGD is actually sub-optimal from a geometric perspective.

- Within the classic DP framework, we propose a geometric perturbation strategy *GeoDP* to directly add the noise on the direction of a gradient, which rigorously guarantees a better trade-off between privacy and efficiency.

- Extensive experiments on public datasets as well as prevalent AI models validate the generality and effectiveness of GeoDP.

The rest of this chapter is organized as follows. Section 5.1 presents our theoretical analysis on deficiency of DP-SGD while Section 5.2 presents the perturbation strategy *GeoDP*. Experimental results are in Section 5.3, followed by a summary in Section 5.4.

Symbol	Meaning
ϵ	privacy budget
β	bounding factor
D	database
S	subset
s	one training data
B	batch size
T	total number of iterations
t	current iteration
C	clipping threshold
σ	noise multiplier
η	learning rate
l	loss function
$\boldsymbol{w}$	model parameters
$\boldsymbol{w}^\star$	global optima
$\boldsymbol{g}$	original gradient
$\tilde{\boldsymbol{g}}$	clipped gradient
$\boldsymbol{n}$	DP noise vector
$\boldsymbol{g}^*$	perturbed gradient from traditional DP
$\boldsymbol{g}^\star$	perturbed gradient from GeoDP
$\boldsymbol{\theta}$	direction of a gradient
$\|\boldsymbol{g}\|$	magnitude of a gradient

Table 5.1: Frequently-used notations

5.1 Deficiency of DP-SGD: a gap between directional SGD and numerical DP

In this section, we identify an intrinsic deficiency in DP-SGD. Let the trained models of DP-SGD and non-private SGD be denoted by $\boldsymbol{w}_{t+1}^* = \boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_t^*$ and $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_t$, respectively. The Euclidean distances between the current models and the global optima (i.e., $\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2$ and $\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2$) reflect the model efficiency of DP-SGD and non-private SGD, respectively. Apparently, the smaller this distance is, the better efficiency the model achieves. Their efficiency difference (ED) (i.e., $\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2$), on the other hand, can describe the impact of DP noise on the model efficiency, as presented by the following theorem.

Theorem 6. *(Impact of DP Noise on Model Efficiency). Suppose $\boldsymbol{n}_\sigma$ follows a noise distribution with the standard deviation $\sigma\boldsymbol{I}$, ED can be measured as:*

$$\begin{aligned}&\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2\\ =&\eta^2\underbrace{\left(\frac{2C}{B}\langle\boldsymbol{n}_\sigma, \tilde{\boldsymbol{g}}_t\rangle + \frac{C^2\boldsymbol{n}_\sigma^2}{B^2}\right)}_{Item\quad A} + \frac{2\eta C}{B}\underbrace{\langle\boldsymbol{n}_\sigma, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle}_{Item\quad B}.\end{aligned} \tag{5.1}$$

Proof. For DP-SGD, we have:

$$\begin{aligned}\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{w}^\star - \eta\tilde{\boldsymbol{g}}_t^*\|^2\\ &= \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 + \eta^2\|\tilde{\boldsymbol{g}}_t^*\|^2 + 2\eta\langle\tilde{\boldsymbol{g}}_t^*, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle.\end{aligned} \tag{5.2}$$

While for SGD, we have:

$$\begin{aligned}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{w}^\star - \eta\tilde{\boldsymbol{g}}_t\|^2\\ &= \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 + \eta^2\|\tilde{\boldsymbol{g}}_t\|^2 + 2\eta\langle\tilde{\boldsymbol{g}}_t, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle.\end{aligned} \tag{5.3}$$

Subtracting Equation 5.3 from Equation 5.2, we have:

$$\begin{aligned}&\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2\\ =&\eta^2\underbrace{\left(\|\tilde{\boldsymbol{g}}_t^*\|^2 - \|\tilde{\boldsymbol{g}}_t\|^2\right)}_{Item\quad A} + 2\eta\underbrace{\langle\tilde{\boldsymbol{g}}_t^* - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle}_{Item\quad B}.\end{aligned} \tag{5.4}$$

Recall that $\boldsymbol{n}_t$ follows a noise distribution whose standard deviation is $C\sigma\boldsymbol{I}$. Suppose $\boldsymbol{n}_\sigma$ follows a noise distribution with the standard deviation $\sigma\boldsymbol{I}$, we have $\boldsymbol{n}_t = C\boldsymbol{n}_\sigma$. For Item A:

$$\begin{aligned}\|\tilde{\boldsymbol{g}}_t^*\|^2 - \|\tilde{\boldsymbol{g}}_t\|^2 &= \left(\tilde{\boldsymbol{g}}_t^* - \tilde{\boldsymbol{g}}_t\right)\left(\tilde{\boldsymbol{g}}_t^* + \tilde{\boldsymbol{g}}_t\right) \\ &= \boldsymbol{n}_t/B\left(2\tilde{\boldsymbol{g}}_t + \boldsymbol{n}_t/B\right) \\ &= 2\langle C\boldsymbol{n}_\sigma/B, \tilde{\boldsymbol{g}}_t\rangle + C^2\boldsymbol{n}_\sigma^2/B^2.\end{aligned} \tag{5.5}$$

And for Item B:

$$\tilde{\boldsymbol{g}}_t^* - \tilde{\boldsymbol{g}}_t = \boldsymbol{n}_t/B = C\boldsymbol{n}_\sigma/B. \tag{5.6}$$

Applying Equation 5.5 and 5.6 into Equation 5.4, we have:

$$\begin{aligned}&\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2 \\ =&\eta^2 \underbrace{\left(2\langle C\boldsymbol{n}_\sigma/B, \tilde{\boldsymbol{g}}_t\rangle + C^2\boldsymbol{n}_\sigma^2/B^2\right)}_{Item\quad A} + 2\eta C/B \underbrace{\langle \boldsymbol{n}_\sigma, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle}_{Item\quad B}.\end{aligned} \tag{5.7}$$

□

In general, we wish the efficiency of DP-SGD closer to SGD, i.e., to make ED as close to zero as possible. This theorem coincides with many empirical findings in existing works. Item A, for example, shows that the introduction of DP noise would cause a bias to the global optima. That is, **DP-SGD cannot stably converges to the global optima, while sometimes reaching that point**, as proved by Corollary 1. This means that the model efficiency of DP-SGD is always lower than regular SGD [24, 125, 142, 164]. In practice, in order to provide a better model efficiency, existing works [1, 42, 159] apply lower noise scale (i.e., smaller $\boldsymbol{n}_\sigma$) when DP-SGD is about to converge. This operation makes Item A close to zero (but normally non-zero). Anther example is that large batch size can enhance the efficiency of DP-SGD, as it can certainly reduce both Item A and Item B [46].

Corollary 1. *DP-SGD cannot stably stays at global optima.*

Proof. Let us just assume DP-SGD reaches the global optima, i.e. $\boldsymbol{w}_t = \boldsymbol{w}^\star$. Accordingly, Item B becomes zero while Item A is non-zero unless $\boldsymbol{n}_\sigma$ stays zero (which

is unlikely), as shown in Equation 5.8. That is, DP noise would immediately cause SGD to deviate from global optima even if SGD can reach optima.

$$\lim_{\boldsymbol{w}_t \to \boldsymbol{w}^\star} \left\|\boldsymbol{w}^*_{t+1} - \boldsymbol{w}^\star\right\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2 = \eta^2 \underbrace{\left(\frac{2C}{B}\langle \boldsymbol{n}_\sigma, \tilde{\boldsymbol{g}}_t\rangle + \frac{C^2 \boldsymbol{n}_\sigma^2}{B^2}\right)}_{Item \quad A}. \tag{5.8}$$

□

More importantly, this theorem reveals that DP-SGD techniques, such as adaptive clipping and learning rate, are incapable of counteracting the impact of DP noise on the direction of a gradient. On one hand, **Item A describes how the noise scale impacts the model efficiency**. To reduce this impact, small learning rate (η^2) and clipping threshold (C and C^2), or large batch size B is effective. This conclusion is confirmed by many existing works, as reviewed in Section 2.2. On the other hand, **Item B**, the inner product between the noise $\boldsymbol{n}_t$ and the training process ($\boldsymbol{w}^\star - \boldsymbol{w}_t$ can be considered as the distance for SGD to descend, i.e., descent trend) **reflects how the perturbation impacts the further training**. While capable of reducing Item A, fine-tuning hyper-parameters cannot reduce Item B, as proved by the following corollary.

Corollary 2. *optimization techniques of DP-SGD (i.e., fine-tuning clipping and learning rate) cannot reduce the impact of noise on the gradient direction.*

Proof. We analyze the effectiveness of DP-SGD techniques (i.e., fine-tuning clipping, learning rate and batch size) on Item A and Item B, respectively.

- *Item A.*

  As per learning rate, we apply different learning rate η^* to DP-SGD, and see if tuning η^* can make Item A zero. Applying η^* to Equation 5.4, we have:

$$\text{Item A} = \eta^{*2}\|\tilde{\boldsymbol{g}}^*_t\|^2 - \eta^2\|\tilde{\boldsymbol{g}}_t\|^2. \tag{5.9}$$

As Equation 5.9 is only composed of numerical values, fined-tuned $\eta^* = \eta^2\|\tilde{\boldsymbol{g}}_t\|^2/\|\tilde{\boldsymbol{g}}_t^*\|^2$ can certainly zero Item A.

As for clipping, given $\boldsymbol{n}_\sigma$ is a random variable drawn from the noise distribution whose standard deviation is $\sigma\boldsymbol{I}$, we have:

$$\boldsymbol{n}_t = C\boldsymbol{n}_\sigma. \tag{5.10}$$

As $\tilde{\boldsymbol{g}}_t^* = \tilde{\boldsymbol{g}}_t + \boldsymbol{n}_t/B$, reducing C certainly reduces the scale of $\tilde{\boldsymbol{g}}_t^*$. Overall, fine-tuning of DP-SGD can certainly reduce Item A.

- *Item B.*

  For learning rate, we have:

$$\begin{aligned} \text{Item B} &= \langle \eta^*\tilde{\boldsymbol{g}}_t^* - \eta\tilde{\boldsymbol{g}}_t, \boldsymbol{w}^\star - \boldsymbol{w}_t\rangle \\ &= \|\eta^*\tilde{\boldsymbol{g}}_t^* - \eta\tilde{\boldsymbol{g}}_t\|\|\boldsymbol{w}^\star - \boldsymbol{w}_t\|\cos\theta. \end{aligned} \tag{5.11}$$

  where θ is the relative angle between two vectors. Apparently, no matter how to fine-tune η^*, how $\eta^*\tilde{\boldsymbol{g}}_t^* - \eta\tilde{\boldsymbol{g}}_t$ varies is rather random because there is no relevance between η^* and $\eta^*\tilde{\boldsymbol{g}}_t^* - \eta\tilde{\boldsymbol{g}}_t$ as well as θ.

  For clipping, we prove that it cannot change the geometric property of the perturbed gradient, although the noise scale is indeed changed. If the clipping thresholds C_1, C_2 and a gradient $\boldsymbol{g}(\|\boldsymbol{g}\| \geq C_1 \geq C_2)$, we have the clipped gradient $\tilde{\boldsymbol{g}}_1 = \frac{\boldsymbol{g}}{\|\boldsymbol{g}_1\|/C_1}$, $\tilde{\boldsymbol{g}}_2 = \frac{\boldsymbol{g}}{\|\boldsymbol{g}_2\|/C_2}$ as per Equation 3.13 and corresponding noise $\boldsymbol{n}_1 = C_1\boldsymbol{n}_\sigma$, $\boldsymbol{n}_2 = C_2\boldsymbol{n}_\sigma$ as per Equation 5.10. Accordingly, the perturbed gradient is:

$$\begin{aligned} \tilde{\boldsymbol{g}}_1^* &= \tilde{\boldsymbol{g}}_1 + \boldsymbol{n}_1/B = \frac{\boldsymbol{g}}{\|\boldsymbol{g}_1\|/C_1} + C_1/B\boldsymbol{n}_\sigma. \\ \tilde{\boldsymbol{g}}_2^* &= \tilde{\boldsymbol{g}}_2 + \boldsymbol{n}_2/B = \frac{\boldsymbol{g}}{\|\boldsymbol{g}_2\|/C_2} + C_2/B\boldsymbol{n}_\sigma. \end{aligned} \tag{5.12}$$

  Then, we have:

$$\begin{aligned} \frac{\tilde{\boldsymbol{g}}_1^*}{C_1} &= \frac{\tilde{\boldsymbol{g}}_2^*}{C_2}. \\ \|\tilde{\boldsymbol{g}}_1^*\| &\geq \|\tilde{\boldsymbol{g}}_2^*\|. \end{aligned} \tag{5.13}$$

Namely, clipping cannot control the directions of perturbed gradients $\frac{\tilde{\boldsymbol{g}}_1^*}{C_1} = \frac{\tilde{\boldsymbol{g}}_2^*}{C_2}$, while indeed reducing the noise scale ($\|\tilde{\boldsymbol{g}}_1^*\| \geq \|\tilde{\boldsymbol{g}}_2^*\|$).

□

In general, this corollary points out an intrinsic deficiency of DP-SGD. That is, as a gradient is actually a vector instead of a numerical array, **traditional DP mechanisms**, which add noise to values of a gradient, **cannot directly reduce the noise on gradient direction (Item B)**. Even worse, **DP introduces biased noise to the direction, while adding unbiased noise to the gradient itself,** as further proved via hyper-spherical coordinate system (see Lemma 6 for rigorous proofs).

5.2 Geometric perturbation: GeoDP

In the previous analysis, we have proved the sub-optimality of traditional DP-SGD. In this section, we seize this opportunity to **perturb the direction and the magnitude of a gradient, respectively, so that the noise on descent trend is directly reduced**. Within the DP framework, our strategy significantly improves the model efficiency.

In what follows, we first introduce d-spherical coordinate system [126] in Section 5.2.1, where one d-dimensional gradient is converted to one magnitude and one direction. By perturbing gradients in the d-spherical coordinate system, we propose our perturbation strategy *GeoDP* to optimize the model efficiency in Section 5.2.2. Privacy and efficiency analysis is provided to prove its compliance with DP definition and huge advantages over DP-SGD in Section 5.2.3.

5.2.1 Hyper-spherical Coordinate System

The d-spherical coordinate system [126], also known as the hyper-spherical coordinate system, is commonly used to analyze geometric objects in high-dimensional space, e.g., the gradient. Compared to the rectangular coordinate system [126], such a system directly represents any d-dimensional vector $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_{d-1}, \boldsymbol{g}_d)$ using a magnitude $\|\boldsymbol{g}\|$ and a direction $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{d-2}, \boldsymbol{\theta}_{d-1})$. Formally, the magnitude is:

$$\|\boldsymbol{g}\| = \sqrt{\sum_{z=1}^{d} \boldsymbol{g}_z^2}. \tag{5.14}$$

and its direction $\boldsymbol{\theta}$ is:

$$\boldsymbol{\theta}_z = \begin{cases} \arctan2\left(\sqrt{\sum_z^{d-1} \boldsymbol{g}_{z+1}^2}, \boldsymbol{g}_z\right) & \text{if } 1 \leq z \leq d-2, \\ \arctan2\left(\boldsymbol{g}_{z+1}, \boldsymbol{g}_z\right) & \text{if } z = d-1. \end{cases}. \tag{5.15}$$

where arctan2 is the two-argument arctangent function defined as follows:

$$\arctan2(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}. \tag{5.16}$$

While having the same functionality as arctan, arctan2 is more robust. For example, arctan2 can deal with a zero denominator ($\boldsymbol{g}_z = 0$). Note that $\sqrt{\sum_z^{d-1} \boldsymbol{g}_{z+1}^2}$ in Equation 5.15 is always non-negative. For $1 \leq z \leq d-2$, the range of $\arctan2\left(\sqrt{\sum_z^{d-1} \boldsymbol{g}_{z+1}^2}, \boldsymbol{g}_z\right)$ is either $\left(0, \frac{\pi}{2}\right]$ or $\left(\frac{\pi}{2}, \pi\right)$ if $\boldsymbol{g}_z \geq 0$ or $\boldsymbol{g}_z < 0$, as per Equation 5.16. **As such, the range of $\boldsymbol{\theta}_{1 \leq z \leq d-2}$ is $(0, \pi)$. For $z = d-1$, the range of $\boldsymbol{\theta}_z$ is $(-\pi, \pi)$ as per Equation 5.16**.

We can also convert a vector $(\|\boldsymbol{g}\|, \boldsymbol{\theta})$ in d-spherical coordinates back to rectangular coordinates $(\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_{d-1}, \boldsymbol{g}_d)$ using the following equation:

$$\boldsymbol{g}_z = \begin{cases} \|\boldsymbol{g}\| \cos \boldsymbol{\theta}_z, & \text{if} \quad z = 1 \\ \|\boldsymbol{g}\| \prod_{i=1}^{z-1} \sin \boldsymbol{\theta}_i \cos \boldsymbol{\theta}_z, & \text{if} \quad 2 \leq z \leq d-1 \\ \|\boldsymbol{g}\| \prod_{i=1}^{z-1} \sin \boldsymbol{\theta}_i, & \text{if} \quad z = d \end{cases} . \tag{5.17}$$

Figure 5.2 provides an example of conversions in three-dimensional space. Given $\|\boldsymbol{g}\| = \sqrt{\boldsymbol{g}_1^2 + \boldsymbol{g}_2^2 + \boldsymbol{g}_3^2}$, $\boldsymbol{\theta}_1 = \text{arctan2}\left(\sqrt{\boldsymbol{g}_2^2 + \boldsymbol{g}_3^2}, \boldsymbol{g}_1\right)$ and $\boldsymbol{\theta}_2 = \text{arctan2}\left(\boldsymbol{g}_3, \boldsymbol{g}_2\right)$, a vector $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3)$ in rectangular coordinate system (marked in black) can be represented as $(\|\boldsymbol{g}\|, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ in hyper-spherical coordinate system (marked in blue). Without loss of generality, we use $\boldsymbol{g} \leftrightarrow (\|\boldsymbol{g}\|, \boldsymbol{\theta})$ to denote the reversible conversions between two systems.

5.2.2 GeoDP—Geometric DP Perturbation for DP-SGD

GeoDP directly reduces the noise on the descent trend via d-spherical coordinate system. Algorithm 5 describes how *GeoDP* works, and major steps are interpreted as follows:

- *Spherical-coordinate Conversion:* Convert the clipped gradient to hyper-spherical coordinate system according to Equation 5.14 and Equation 5.15, i.e., $\boldsymbol{g} \rightarrow (\|\boldsymbol{g}\|, \boldsymbol{\theta})$ ($\rightarrow$ means coordinate conversion), which allows perturbation on the magnitude and the direction of a gradient, respectively.

- *Reducing the Direction Range (Sensitivity):* According to Theorem 9, the averaged direction of gradients $\left\{\tilde{\boldsymbol{g}}_{tj} | 1 \leq j \leq B\right\}$ should be centered at one small range, rather than uniformly spreading the whole vector space. This conclusion is also confirmed by various SGD studies [15, 159]. DP-SGD, taking the whole direction space as the privacy region, is therefore overprotective and low

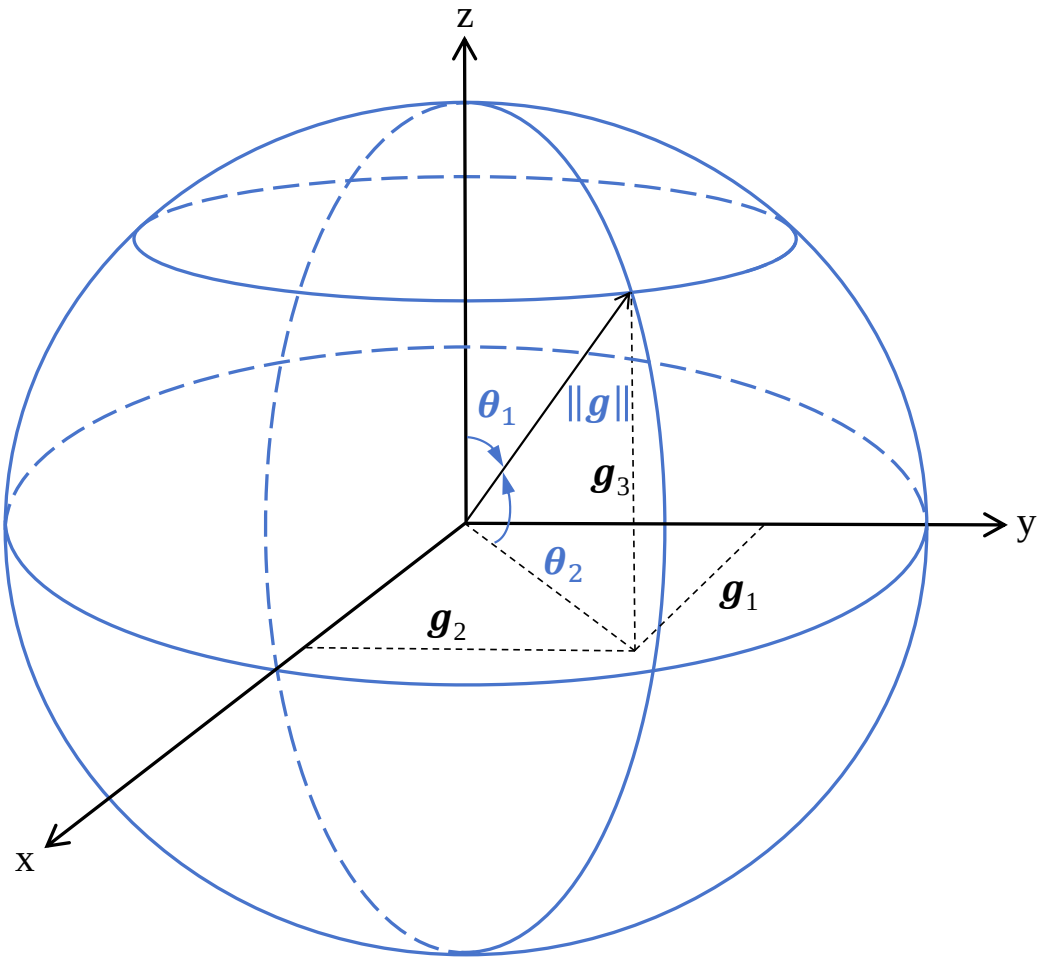


Figure 5.2: Coordinates Conversions in Three-dimensional Space.

efficient. In this work, a bounding factor $\beta \in (0, 1]$ defines the privacy region into a subspace around the original direction, which significantly reduces the noise addition in Step 3. For $1 \leq z < d-1$, given $0 \leq \Gamma_1 \leq \boldsymbol{\theta}_z \leq \Gamma_2 \leq \pi$, β determines the range between Γ_1 and Γ_2, i.e., $\Gamma_2 - \Gamma_1 = \Delta\boldsymbol{\theta}_z = \beta\pi$. Similarly, $\Gamma_2 - \Gamma_1 = \Delta\boldsymbol{\theta}_z = 2\beta\pi$ for $z = d-1$. Note that $\beta = 1$ means the full space. This parameter directly determines the sensitivity of the direction, which consequently influences the noise addition in the following step. because $\boldsymbol{\theta}$ is essentially an array instead of a vector. Under this factor, this stage clips directions into $\tilde{\boldsymbol{\theta}}_{1\leq z\leq d-2} \in ((1-\beta)\pi, \pi)$ while $\tilde{\boldsymbol{\theta}}_{d-1} \in (-\beta\pi, \beta\pi)$. Besides, in terms of $\frac{\sqrt{\sum_z^{d-1} \boldsymbol{g}_{z+1}^2}}{\boldsymbol{g}_z}$, we observe that the numerator is normally larger than the denominator. That is, $\arctan2\left(\sqrt{\sum_z^{d-1} \boldsymbol{g}_{z+1}^2}, \boldsymbol{g}_z\right)$ is more probably located on the right side of $(0, \pi)$. As such, we prefer the range $((1-\beta)\pi, \pi)$ for $\tilde{\boldsymbol{\theta}}_{1\leq z\leq d-2}$.

- *Noise Addition:* GeoDP allows to perturb the magnitude and the direction of a gradient, respectively. For the magnitude, $\|\tilde{\boldsymbol{g}}_t\|$ is already bounded by C in the first stage. Similar to DP-SGD, the noise scale of the perturbed magnitude is $C\sigma$. For the direction, the noise scale is the sensitivity $\Delta\boldsymbol{\theta}$ times the noise multiplier σ. Note that maximum changes of $\tilde{\boldsymbol{\theta}}_{1\leq z\leq d-2}$ and $\tilde{\boldsymbol{\theta}}_{d-1}$ are $\beta\pi$ and $2\beta\pi$, respectively, due to the bounding of the direction range. Overall, $\Delta\boldsymbol{\theta} = \sqrt{(d-2)(\beta\pi)^2 + (2\beta\pi)^2} = \sqrt{d+2}\beta\pi$.

- *Rectangular-coordinate Conversion:* Convert the perturbed magnitude and direction back to rectangular coordinates according to Equation 5.17, i.e., $(\|\tilde{\boldsymbol{g}}_t\|^\star, \boldsymbol{\theta}_t^\star) \rightarrow \tilde{\boldsymbol{g}}_t^\star$, which allows future gradient descent.

In general, GeoDP provides better efficiency to SGD in two perspectives. First, **GeoDP adds unbiased noise, whereas traditional DP introduces biased perturbation, to the direction of a gradient** (see Lemma 6 for rigorous proofs). This counter-intuitive conclusion is supported by the fact that tradition DP, which adds unbiased noise to the gradient itself, however accumulates noise on different an-

Algorithm 5 GeoDP-SGD

Input: Batch size B, noise multiplier σ, clipping threshold C, bounding factor $\beta(0 < \beta \leq 1)$, learning rate η, total number of iterations T.

Output: Trained model $\boldsymbol{w}_T^\star$.

1: Initialize a model with parameters $\boldsymbol{w}_0$.

2: **for** each iteration $t = 0, 1, ..., T-2, T-1$ **do**

3: Derive the average clipped gradient $\tilde{\boldsymbol{g}}_t$ with respect to the batch size B and the clipping threshold C.

4: Convert $\tilde{\boldsymbol{g}}_t$ to d-spherical coordinates as $(\|\tilde{\boldsymbol{g}}_t\|, \boldsymbol{\theta}_t)$.

5: Clip $\boldsymbol{\theta}_t$ into $\tilde{\boldsymbol{\theta}}$ as follows:

$$\tilde{\boldsymbol{\theta}} = \begin{cases} \tilde{\boldsymbol{\theta}}_{1\leq z\leq d-2} & = \begin{cases} \boldsymbol{\theta}_z & \text{if} \quad \boldsymbol{\theta}_z > (1-\beta)\pi, \\ (1-\beta)\pi & \text{if} \quad \boldsymbol{\theta}_z \leq (1-\beta)\pi. \end{cases} \\ \tilde{\boldsymbol{\theta}}_{d-1} & = \begin{cases} \boldsymbol{\theta}_z & \text{if} \quad \|\boldsymbol{\theta}_z\| < \beta\pi, \\ \beta\pi & \text{if} \quad \boldsymbol{\theta}_z \geq \beta\pi, \\ -\beta\pi & \text{if} \quad \boldsymbol{\theta}_z \leq -\beta\pi. \end{cases} \end{cases}$$

Bound the privacy region Δ of $\boldsymbol{\theta}$ as follows:

$$\Delta\boldsymbol{\theta}_z = \begin{cases} \Delta\boldsymbol{\theta}_{1\leq z\leq d-2} & = \beta\pi, \\ \Delta\boldsymbol{\theta}_{d-1} & = 2\beta\pi. \end{cases}$$

6: $\|\tilde{\boldsymbol{g}}_t\|^\star = \|\tilde{\boldsymbol{g}}_t\| + \frac{C}{B}\boldsymbol{n}_\sigma$, $\tilde{\boldsymbol{\theta}}_t^\star = \tilde{\boldsymbol{\theta}}_t + \frac{\sqrt{d+2}\beta\pi}{B}\boldsymbol{n}_\sigma$, where $\boldsymbol{n}_\sigma$ follows a zero-mean Gaussian distribution with standard deviation σ.

7: Convert $\left(\|\tilde{\boldsymbol{g}}_t\|^\star, \tilde{\boldsymbol{\theta}}_t^\star\right)$ back to rectangular coordinates as the perturbed gradient $\tilde{\boldsymbol{g}}_t^\star$.

8: Update $\boldsymbol{w}_{t+1}^\star$ by taking a step in the direction of the noisy gradient, i.e., $\boldsymbol{w}_{t+1}^\star = \boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_t^\star$.

9: **end for**

gles of one direction. Example 2 demonstrates how this noise accumulation happens. As such, numerical perturbation of DP seriously degrades the accuracy of directional information. GeoDP, on the other hand, independently controls the noise on each angle and therefore prevents noise accumulation.

Example 2. *Suppose we have a three-dimensional gradient* $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3)$. *Following traditional DP, these three should be added noise* $\boldsymbol{n} = (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)$. *For the direction of this perturbed gradient* $\boldsymbol{\theta}$, *according to Equation 4, its first angle* $\boldsymbol{\theta}_1$ *should be* $\arctan2\left(\sqrt{(\boldsymbol{g}_2+\boldsymbol{n}_2)^2+(\boldsymbol{g}_3+\boldsymbol{n}_3)^2}, \boldsymbol{g}_1+\boldsymbol{n}_1\right)$. *It is very obvious that noise of three dimensions* $(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)$ *is accumulated to the first angle* $\boldsymbol{\theta}_1$, *and this accumulation is biased.*

Second, via coordinates conversion, d-dimensional gradient is transferred to one magnitude and $d-1$ directions. By composition theory, $\frac{d-1}{d}$ privacy budget is allocated to the direction by GeoDP, which can better preserves directional information.

Theorem 7. *(Privacy Cost of GeoDP in (α,ϵ)-RDP). Given* $\tilde{\boldsymbol{g}} \leftrightarrow \left(\|\tilde{\boldsymbol{g}}\|, \tilde{\boldsymbol{\theta}}\right)$, $\tilde{\boldsymbol{g}}^\star$ *satisfies* $\left(\alpha, \frac{\epsilon_1+(d-1)\epsilon_2}{d}\right)$*-DP if* $\|\tilde{\boldsymbol{g}}\|^\star$ *and* $\tilde{\boldsymbol{\theta}}^\star$ *follow* (α, ϵ_1)*- and* (α, ϵ_2)*-DP, respectively.*

Proof. Given two neighboring dataset D, D' and their output sets $(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^\star) = \left\{(\tilde{\boldsymbol{g}}_1^\star, \tilde{\boldsymbol{\theta}}_1^\star), ...\right\}$ of $\mathcal{M}(D)$, $(\tilde{\boldsymbol{g}}^{\star'}, \tilde{\boldsymbol{\theta}}^{\star'}) = \left\{(\tilde{\boldsymbol{g}}_1^{\star'}, \tilde{\boldsymbol{\theta}}_1^{\star'}), ...\right\}$ of $\mathcal{M}(D')$, respectively, we have:

$$
\begin{aligned}
& D_\alpha(\mathcal{M}(D)||\mathcal{M}(D')) \\
=& \frac{1}{\alpha-1}\log \mathbb{E}_{x\sim\mathcal{M}(D)}\left[\left(\frac{\Pr\left[\mathcal{M}(D)=x\right]}{\Pr\left[\mathcal{M}(D')=x\right]}\right)^{\alpha-1}\right] \\
=& \frac{1}{\alpha-1}\log \mathbb{E}_{(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^\star)\to\mathcal{M}(D)}\left[\left(\frac{\Pr\left[(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^\star)\to x\right]}{\Pr\left[(\tilde{\boldsymbol{g}}^{\star'}, \tilde{\boldsymbol{\theta}}^{\star'})\to x\right]}\right)^{\alpha-1}\right].
\end{aligned}
\tag{5.18}
$$

Because $\|\tilde{\boldsymbol{g}}\|$ only has one dimension while $\tilde{\boldsymbol{\theta}}$ has $d-1$ dimensions (d dimensions in

total), we have:

$$\begin{aligned}
&\log \mathbb{E}_{(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to\mathcal{M}(D)}\left[\left(\frac{\Pr\left[(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to x\right]}{\Pr\left[(\tilde{\boldsymbol{g}}^{\star'},\tilde{\boldsymbol{\theta}}^{\star'})\to x\right]}\right)^{\alpha-1}\right] \\
\leq &\frac{1}{d}\log \mathbb{E}_{(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to\mathcal{M}(D)}\left[\left(\frac{\Pr\left[(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to x\right]}{\Pr\left[(\tilde{\boldsymbol{g}}^{\star'},\tilde{\boldsymbol{\theta}}^\star)\to x\right]}\right)^{\alpha}\right] \\
&+\frac{d-1}{d}\log \mathbb{E}_{(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to\mathcal{M}(D)}\left[\left(\frac{\Pr\left[(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^\star)\to x\right]}{\Pr\left[(\tilde{\boldsymbol{g}}^\star,\tilde{\boldsymbol{\theta}}^{\star'})\to x\right]}\right)^{\alpha}\right].
\end{aligned} \tag{5.19}$$

Applying Equation 5.19 to Equation 5.18, we have:

$$D_\alpha(\mathcal{M}(D)||\mathcal{M}(D')) \leq \frac{\epsilon_1}{d}+\frac{(d-1)\epsilon_2}{d}=\frac{\epsilon_1+(d-1)\epsilon_2}{d}. \tag{5.20}$$

by which this theorem is proven. □

Remark 1. *If both $\|\tilde{\boldsymbol{g}}\|^\star$ and $\tilde{\boldsymbol{\theta}}^\star$ follow (α,ϵ)-DP, $\boldsymbol{g}^\star$ follows (α,ϵ)-RDP.*

Finally, we discuss the time complexity of GeoDP-SGD. For DP-SGD, given the size of private dataset $|D|$ and the number of gradient's dimensions d, DP-SGD takes $O(|D|d)$ time to calculate derivatives in one epoch [159]. By contrast, coordinate conversion costs a little time because it only involves simple geometry calculation. Besides that, GeoDP has the same time complexity as DP-SGD.

5.2.3 Comparison between GeoDP and Traditional DP: Efficiency and Privacy

5.2.3.1 Efficiency Comparison

Via hyper-spherical coordinate system, we can identify deficiencies of traditional DP from a geometric perspective and further understand the merits of GeoDP. If clipping threshold is fixed, the max magnitude of a clipped gradient is determined, because

$\|\tilde{\boldsymbol{g}}\| = \frac{\|\tilde{\boldsymbol{g}}\|}{\max\{1, \|\boldsymbol{g}\|/C\}} \leq C$. That is, the clipped gradients are within the hyper-sphere whose radius (abbreviated as R) is C. Figure 5.2 can help to understand this fact. For example, $\boldsymbol{g}$ (highlighted in black) in Figure 5.2 is vector within the hyper-sphere whose radius is $\|\boldsymbol{g}\|$ (highlighted in blue). By adding noise, traditional DP makes sure that any two gradients within the hyper-sphere are indistinguishable. However, there are two serious disadvantages.

On one hand, numerical noise addition does not respect the geometric property of gradients, as interpreted by the following example. In general, traditional DP seriously sabotages the geometric property of a gradient, which eventually results in low model efficiency.

Example 3. *Suppose two parallel gradients* $\tilde{\boldsymbol{g}}_1 = (1, 1)$, $\tilde{\boldsymbol{g}}_2 = (2, 2)$ *and clipping threshold* $C = 2\sqrt{2}$. *As such, these two gradients are all within* $R = C = 2\sqrt{2}$ *hyper-sphere, and their directions are all* $\boldsymbol{\theta} = \text{arctan2}(1, 1) = \text{arctan2}(2, 2) = \frac{\pi}{4}$. *As such, DP adds the same scale of noise to both gradients for privacy preservation. Assuming that the noise* $\boldsymbol{n} = (2, -1)$ *is added to both gradients, directions of two perturbed gradients are* $\boldsymbol{\theta}_1^* = \text{arctan2}(1 - 1, 1 + 2) = 0$ *and* $\boldsymbol{\theta}_2^* = \text{arctan2}(2 - 1, 2 + 2) \approx \frac{2\pi}{25}$. *Given parallel gradients (*$\boldsymbol{\theta} = \frac{\pi}{4}$*), directions of perturbed gradients (*$\boldsymbol{\theta}_1^* \neq \boldsymbol{\theta}_2^* \neq \boldsymbol{\theta}$*) are much different, even if the added noise (*$\boldsymbol{n} = (2, -1)$*) is the same.*

On the other hand, traditional DP, which preserves all directions within the hyper-sphere, actually adds excessive noise to the gradient. Different from regular SGD, DP-SGD usually requires very large batch size (e.g., 16,384) to reduce the negative impact of noise [46], which makes training process less "stochastic" [15, 159]. In specific, the summation of gradients $\left\{\tilde{\boldsymbol{g}}_{jz} | 1 \leq j \leq B, 1 \leq z \leq d\right\}$ follows *Lindeberg–Lévy Central Limit Theorem* (CLT) [115] as these gradients are independently and identically distributed (each of them is derived from a single data of the same dataset). As such, we can use Gaussian distribution to model the average of this summation (i.e., $\tilde{\boldsymbol{g}}_z = \frac{1}{B}\sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{jz}$), as proved by the following theorem.

Theorem 8. *(Modeling of the Averaged Stochastic Gradients). Suppose that* $var(\tilde{\boldsymbol{g}}_{jz})$ *and* $\mathbb{E}(\tilde{\boldsymbol{g}}_{jz})$ *are the variance and the expectation of* $\left\{\tilde{\boldsymbol{g}}_{jz}|1 \leq j \leq B, 1 \leq z \leq d\right\}$*, the probability density function (pdf) of* $\tilde{\boldsymbol{g}}_z$ *is:*

$$\lim_{B\to\infty} \tilde{\boldsymbol{g}} \sim \mathcal{N}\left(\mathbb{E}(\tilde{\boldsymbol{g}}_j), \sqrt{\frac{var(\tilde{\boldsymbol{g}}_j)}{B}}\right) \lim_{B\to\infty} f(\tilde{\boldsymbol{g}}_z) = \sqrt{\frac{B}{2\pi * var(\tilde{\boldsymbol{g}}_{jz})}} \exp\left(-\frac{B^2 * \left(x - \mathbb{E}(\tilde{\boldsymbol{g}}_{jz})\right)^2}{2 * var(\tilde{\boldsymbol{g}}_{jz})}\right). \tag{5.21}$$

Proof. $\left\{\tilde{\boldsymbol{g}}_j|1 \leq j \leq B\right\}$ are independently and identically distributed variables because each one is derived from one data s_j of the same subset S. According to *CLT*, the following probability holds:

$$\begin{aligned} &\lim_{B\to\infty} \Pr\left(\frac{\sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{jz} - B * \mathbb{E}(\tilde{\boldsymbol{g}}_{jz})}{\sqrt{B * var(\tilde{\boldsymbol{g}}_{jz})}} \leq X\right) \\ = &\lim_{B\to\infty} \Pr\left(\frac{\frac{1}{B}\sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{jz} - \mathbb{E}(\tilde{\boldsymbol{g}}_{jz})}{\sqrt{var(\tilde{\boldsymbol{g}}_{jz})/B}} \leq X\right) = \int_{-\infty}^{X} \phi(x)dx. \end{aligned} \tag{5.22}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$ is the pdf of the standard Gaussian distribution. As such, $\frac{\sum_{j=1}^{B} \tilde{\boldsymbol{g}}_{jz}/B - \mathbb{E}(\tilde{\boldsymbol{g}}_{jz})}{\sqrt{var(\tilde{\boldsymbol{g}}_{jz})/B}}$ follows standard gaussian distribution $\mathcal{N}(0, 1)$, by which our claim is proved. □

Indicated by Theorem 8, large batch size would incur unevenly distributed average of gradients, making the training process less stochastic. A further conjecture proposes that some directions within the space are also unlikely to be the direction of gradient descent at the current state, as proved by the following theorem. Suppose that the directions of all gradients are $\{\boldsymbol{\theta}_{jz}|1 \leq j \leq B, 1 \leq z \leq d\}$, we have:

Theorem 9. *(Modeling of the Averaged Directions of Gradients). Suppose that* $var(\tilde{\boldsymbol{\theta}}_{jz})$ *and* $\mathbb{E}(\tilde{\boldsymbol{\theta}}_{jz})$ *are the variance and expectation of* $\left\{\tilde{\boldsymbol{\theta}}_{jz}|1 \leq j \leq B, 1 \leq z \leq d\right\}$*,*

the pdf of the averaged direction $\tilde{\boldsymbol{\theta}}_z = \frac{1}{B}\sum_{j=1}^{B}\tilde{\boldsymbol{\theta}}_{jz}$ *is:*

$$\lim_{B\to\infty}\boldsymbol{\theta} \sim \mathcal{N}\left(\mathbb{E}(\boldsymbol{\theta}_j), \sqrt{\frac{var(\boldsymbol{\theta}_j)}{B}}\right) \lim_{B\to\infty} f(\tilde{\boldsymbol{\theta}}_z) = \sqrt{\frac{B}{2\pi * var(\tilde{\boldsymbol{\theta}}_{jz})}}\exp\left(-\frac{B^2 * \left(x - \mathbb{E}(\tilde{\boldsymbol{\theta}}_{jz})\right)^2}{2 * var(\tilde{\boldsymbol{\theta}}_{jz})}\right). \tag{5.23}$$

Proof.

$$\begin{aligned}&\lim_{B\to\infty}\Pr\left(\frac{\sum_{j=1}^{B}\tilde{\boldsymbol{\theta}}_j - B * \mathbb{E}(\tilde{\boldsymbol{\theta}}_j)}{\sqrt{B * var(\tilde{\boldsymbol{\theta}}_j)}} \le X\right)\\ =&\lim_{B\to\infty}\Pr\left(\frac{\frac{1}{B}\sum_{j=1}^{B}\tilde{\boldsymbol{\theta}}_j - \mathbb{E}(\tilde{\boldsymbol{\theta}}_j)}{\sqrt{var(\tilde{\boldsymbol{\theta}}_j)/B}} \le X\right) = \int_{-\infty}^{X}\phi(x)dx.\end{aligned} \tag{5.24}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})$ is the pdf of the standard Gaussian distribution. As such, $\frac{\sum_{j=1}^{B}\boldsymbol{\theta}_j/B - \mathbb{E}(\boldsymbol{\theta}_j)}{\sqrt{var(\boldsymbol{\theta}_j)/B}}$ follows standard gaussian distribution $\mathcal{N}(0,1)$, by which our claim is proved. □

This theorem proves that the averaged direction of stochastic gradients actually concentrated at a certain direction, rather than spreading in the whole vector space. As such, traditional DP-SGD, only effective in the whole vector space, actually wastes privacy budgets to preserve unnecessary directions. In contrast, GeoDP preserves the subspace where directions of various gradients are concentrated, and therefore provides much better efficiency, as jointly proved by the following lemma (which indicates the better accuracy of GeoDP on preserving directional information) and theorem (which further indicates the superiority of GeoDP on model efficiency). Experimental results in Section 5.3.2 also confirm our analysis.

Lemma 6. *Given the original direction* $\boldsymbol{\theta}$*, two perturbed directions* $\boldsymbol{\theta}^\star$ *and* $\boldsymbol{\theta}^*$ *from GeoDP and DP, respectively, there always exists such a bounding factor* β *that MSE*$(\tilde{\boldsymbol{\theta}}_t^\star) <$ *MSE*$(\tilde{\boldsymbol{\theta}}_t^*)$ *holds.*

Proof. For traditional DP (adding noise $\boldsymbol{n}$ to the gradient $\boldsymbol{g}$), we can derive the perturbed angle $\boldsymbol{\theta}_z^*$ according to Equation 5.15, i.e.,

$$\boldsymbol{\theta}_z^* = \begin{cases} \arctan2\left(\sqrt{\sum_z^{d-1}(\boldsymbol{g}_{z+1}+\boldsymbol{n}_{z+1})^2}, \boldsymbol{g}_z+\boldsymbol{n}_z\right) & \text{if } 1 \le z \le d-2, \\ \arctan2\left(\boldsymbol{g}_{z+1}+\boldsymbol{n}_{z+1}, \boldsymbol{g}_z+\boldsymbol{n}_z\right) & \text{if } z = d-1. \end{cases} . \tag{5.25}$$

Observing both acrtan2 equations above, we can conclude that the **traditional DP perturbation** introduces **biased** noise to the original direction, i.e., $\mathbb{E}(\boldsymbol{\theta}^*) \neq \boldsymbol{\theta}(bias(\boldsymbol{\theta}^*) \neq 0)$. Also, the variance of $\boldsymbol{\theta}$ $(var(\boldsymbol{\theta}^*))$ is non-zero, if the noise scale $\boldsymbol{n}_\sigma > 0$.

For GeoDP, we have $\boldsymbol{\theta}^\star = \boldsymbol{\theta} + \frac{\sqrt{d+2}\beta\pi}{B}\boldsymbol{n}_\sigma$. Accordingly, $\mathbb{E}(\boldsymbol{\theta}^\star) = \mathbb{E}(\boldsymbol{\theta} + \frac{\sqrt{d+2}\beta\pi}{B}\boldsymbol{n}_\sigma) = \boldsymbol{\theta}(bias(\boldsymbol{\theta}^\star) = 0)$, which means that GeoDP adds unbiased noise to the direction. Besides, *beta* directly controls the noise added to the direction. In specific, the variance of $\boldsymbol{\theta}^\star(var(\boldsymbol{\theta}^\star))$ can approaching zero if $\beta \to 0$, because $\boldsymbol{\theta}^\star = \boldsymbol{\theta} + \frac{\sqrt{d+2}\beta\pi}{B}\boldsymbol{n}_\sigma$ approaches 0 if $\beta \to 0$.

Given that $\text{MSE}(\boldsymbol{\theta}) = bias^2(\boldsymbol{\theta}) + var(\boldsymbol{\theta})$ [31], there always exist such one β that:

$$\text{MSE}(\boldsymbol{\theta}^\star) = bias^2(\boldsymbol{\theta}^\star) + var(\boldsymbol{\theta}^\star) <= bias^2(\boldsymbol{\theta}^*) + var(\boldsymbol{\theta}^*) = \text{MSE}(\boldsymbol{\theta}^*). \tag{5.26}$$

by which our claim is proven. □

Supported by this lemma, we further prove the optimality of GeoDP to tradition DP in the efficiency of SGD tasks in the next theorem.

Theorem 10. *(Optimality of GeoDP). Let* $\boldsymbol{w}_{t+1}^\star = \boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_t^\star$, $\boldsymbol{w}_{t+1}^* = \boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_t^*$ *and* $\tilde{\boldsymbol{g}}_t$, $\tilde{\boldsymbol{g}}_t^\star$ *and* $\tilde{\boldsymbol{g}}_t^*$ *be the clipped gradient, noisy gradients of GeoDP and DP, respectively. Besides,* $\tilde{\boldsymbol{g}}_t \to \left(\|\tilde{\boldsymbol{g}}_t\|, \tilde{\boldsymbol{\theta}}_t\right)$, $\tilde{\boldsymbol{g}}_t^\star \to \left(\|\tilde{\boldsymbol{g}}_t\|^\star, \tilde{\boldsymbol{\theta}}_t^\star\right)$ *and* $\tilde{\boldsymbol{g}}_t^* \to \left(\|\tilde{\boldsymbol{g}}_t\|^*, \tilde{\boldsymbol{\theta}}_t^*\right)$. *The following inequality always holds if* $\tilde{\boldsymbol{g}}_t^\star$ *and* $\tilde{\boldsymbol{g}}_t^*$ *both follow* (ϵ, δ)*-DP:*

$$\mathbb{E}\left(\left\|\boldsymbol{w}_{t+1}^\star - \boldsymbol{w}^\star\right\|^2\right) < \mathbb{E}\left(\left\|\boldsymbol{w}_{t+1}^* - \boldsymbol{w}^\star\right\|^2\right). \tag{5.27}$$

Proof. Following Corollary 2, we just have to prove Item B of GeoDP is smaller than Item A of DP. Different learning rates $\eta^\star$ and η^* are applied to GeoDP and DP, respectively. Recall from Corollary 2, we have:

$$\begin{aligned} \text{Item B} &= \langle \eta^\star \tilde{\boldsymbol{g}}_t^\star - \eta \tilde{\boldsymbol{g}}_t, \boldsymbol{w}^\star - \boldsymbol{w}_t \rangle \\ &= \underbrace{\|\eta^\star \tilde{\boldsymbol{g}}_t^\star - \eta \tilde{\boldsymbol{g}}_t\|}_{C} \underbrace{\|\boldsymbol{w}^\star - \boldsymbol{w}_t\|}_{D} \underbrace{\cos\theta}_{E}. \end{aligned} \tag{5.28}$$

Note that the only way to optimize Item B is via Item C. Most likely, Item D, as the distance between the current model and the optima, is fixed, and Item E, which describes the relative angle between noise and the fixed distance, is too random to handle. Therefore, we manage to zero Item C as much as possible to optimize Item B. In general, we have:

$$\text{Item C}^2 = (\eta^\star \tilde{\boldsymbol{g}}_t^\star)^2 + (\eta \tilde{\boldsymbol{g}}_t)^2 - 2\eta^\star \eta \langle \tilde{\boldsymbol{g}}_t^\star, \tilde{\boldsymbol{g}}_t \rangle. \tag{5.29}$$

While $(\eta^\star \tilde{\boldsymbol{g}}_t^\star)^2 + (\eta \tilde{\boldsymbol{g}}_t)^2$ can be fine-tuned to zero by learning rates, the only way for $\langle \tilde{\boldsymbol{g}}_t^\star, \tilde{\boldsymbol{g}}_t \rangle$ to be zero is that the direction of $\boldsymbol{g}^\star$ should approximate that of $\tilde{\boldsymbol{g}}_t$ (or the opposite direction of $\tilde{\boldsymbol{g}}_t$, which rarely happens and is therefore out of question here.). Due to $\text{MSE}(\tilde{\boldsymbol{\theta}}_t^\star) < \text{MSE}(\tilde{\boldsymbol{\theta}}_t^*)$ in Lemma 6, GeoDP can therefore more easily make Item B zero than DP, by which our claim is proved. □

5.2.3.2 Privacy Comparison

Now that the superiority of GeoDP on model efficiency is rigorously analyzed, we next prove its alignment with the formal DP definition. The following lemma and theorem analyze the privacy level of perturbed gradient direction and gradient itself of GeoDP, respectively.

Lemma 7. *The perturbed direction from GeoDP $\tilde{\boldsymbol{\theta}}^\star$ under β bounding factor satisfies $(\epsilon, \delta + \delta')-DP$, where*

$$1 - \int_0^{2\beta\pi} \underbrace{\int_0^{\beta\pi} \dots \int_0^{\beta\pi}}_{d-1} \prod_{z=1}^{d} f(\tilde{\boldsymbol{\theta}}_z) d\tilde{\boldsymbol{\theta}}_z \leq \delta' \leq 1 - \beta. \tag{5.30}$$

Proof. While δ covers the probability where the strict DP is ineffective [34, 36, 37], we use δ' to denote the probability of space where (ϵ, δ)-DP is ineffective. Since $\tilde{\boldsymbol{\theta}}^\star$ is generally not the expectation of $\{\boldsymbol{\theta}_j\}$, we have:

$$\delta' \geq 1 - \int_0^{2\beta\pi} \underbrace{\int_0^{\beta\pi} ... \int_0^{\beta\pi}}_{d-1} \prod_{z=1}^{d} f(\tilde{\boldsymbol{\theta}}_z) d\tilde{\boldsymbol{\theta}}_z. \tag{5.31}$$

Meanwhile, the space that β cannot cover is $1 - \beta$ if the directions are evenly distributed (as discussed before, they are not). As such, $\delta' \leq 1 - \beta$, by which our claim is proved. □

Theorem 11. *(Privacy Level of GeoDP). Given $\tilde{\boldsymbol{g}} \leftrightarrow \left(\|\tilde{\boldsymbol{g}}\|, \tilde{\boldsymbol{\theta}}\right)$, $\tilde{\boldsymbol{g}}^\star$ satisfies $(\epsilon, \delta + \delta')$-DP if $\|\tilde{\boldsymbol{g}}\|^\star$ and $\tilde{\boldsymbol{\theta}}^\star$ follow (ϵ, δ)-DP and $(\epsilon, \delta + \delta')$-DP, respectively.*

Proof. Given two neighboring dataset D, D' and their output sets $(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^\star) = \left\{(\tilde{\boldsymbol{g}}_1^\star, \tilde{\boldsymbol{\theta}}_1^\star), ...\right\}$ of $\mathcal{M}(D)$, $(\tilde{\boldsymbol{g}}^{\star'}, \tilde{\boldsymbol{\theta}}^{\star'}) = \left\{(\tilde{\boldsymbol{g}}_1^{\star'}, \tilde{\boldsymbol{\theta}}_1^{\star'}), ...\right\}$ of $\mathcal{M}(D')$, respectively, we have:

$$\begin{aligned}
&\Pr[\mathcal{M}(D) \in S] = \Pr[(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^\star) \in S] \\
\leq &\left(e^\epsilon \Pr[(\tilde{\boldsymbol{g}}^{\star'}, \tilde{\boldsymbol{\theta}}^\star) \in S] + \delta\right) \vee \left(e^\epsilon \Pr[(\tilde{\boldsymbol{g}}^\star, \tilde{\boldsymbol{\theta}}^{\star'}) \in S] + \delta + \delta'\right) \\
= &\left(e^\epsilon \Pr[(\tilde{\boldsymbol{g}}^{\star'}, \tilde{\boldsymbol{\theta}}^{\star'}) \in S] + \delta + \delta'\right) \\
= &e^\epsilon \Pr[\mathcal{M}(D') \in S] + \delta + \delta'.
\end{aligned} \tag{5.32}$$

by which this theorem is proven. □

Compared with traditional DP which imposes (ϵ, δ)-DP on the whole gradient, GeoDP relieves the privacy level of gradient direction (i.e., $\tilde{\boldsymbol{\theta}}^\star$ satisfies $(\epsilon, \delta + \delta')$-DP) while maintaining the same privacy preservation on gradient magnitude (i.e., $\tilde{\boldsymbol{g}}^\star$ satisfies (ϵ, δ)-DP). In return, the model efficiency of SGD is much improved under the same noise scale. While the privacy preservation is weaker, GeoDP imposes more perturbation on gradient magnitude, making it even harder for various attacks to succeed.

5.3 Experimental results

This section empirically evaluates our analysis as well as the perturbation strategy *GeoDP* in various learning tasks. In Section 5.3.2, we first validate Lemma 6 where GeoDP preserves directional information better than traditional DP. We then compare performances of GeoDP with traditional DP in one machine learning model (i.e., Logistic Regression) and two deep learning models (i.e., CNN and ResNet) in Section 5.3.3 and Section 5.3.4, respectively. Since GeoDP only modifies the way to perturb, instead of the target to be perturbed and the training process, existing optimization techniques, such as adaptive clipping and other advanced optimizers, are orthogonal to GeoDP. To demonstrate generality of GeoDP, we also compare the performance of GeoDP and DP in CNN with a state-of-the-art clipping technique AUTO-S [17]. Finally, to evaluate the practical privacy risk, we implement a benchmark membership inference attack (MIA) on both GeoDP and DP.

5.3.1 Experimental Setup

We conduct our experiments on a server with Intel Xeon Silver 4210R CPU, 128G RAM, and Nvidia GeForce RTX 3090 GPU on Ubuntu 20.04 LTS system. All results are repeated 100 times to obtain the average. Unless otherwise specified, we fix $C = 0.1$.

5.3.1.1 Datasets and Models

For model efficiency, we use two prevalent benchmark datasets, MNIST [76] and CIFAR-10 [75]. For MIA tests, we adopt the state-of-the-art benchmark ML-DOCTOR [118] and use four public datasets, MNIST, CIFAR-10, CelebA [89], and FMNIST [143]. Besides, we also conduct a standalone experiment to verify that GeoDP preserves directional information better than DP (Lemma 6). Due to the lack of public gradient

datasets, we form a synthetic one for this experiment. The details of these datasets are as below.

MNIST. This is a dataset of 70,000 gray-scale images (28x28 pixels) of handwritten digits from 0 to 9, commonly used for training and testing machine learning algorithms in image recognition tasks. It consists of 60,000 training images and 10,000 testing images, with an even distribution across the 10 digit classes.

CIFAR-10. It is a dataset of 60,000 small (32x32 pixels) color images, divided into 10 distinct classes such as animals and vehicles, used for machine learning and computer vision tasks. It contains 50,000 training images and 10,000 testing images, with each class having an equal number of images.

Synthetic Gradient Dataset. To synthesize a dataset of gradients, we randomly collect $450,000$ gradients (of $20,000$ dimensions) from 9 epochs of training a non-DP CNN ($B = 1$) on CIFAR-10 (i.e., $50,000$ training images). Dimensions are randomly chosen in various experiments.

As for models, recall that our experiments aim to confirm the superiority of GeoDP to DP on SGD, instead of yearning the best empirical accuracy over all existing ML models. As such, we believe prevalent models such as LR, 2-layer CNN with Softmax activation and ResNet with 3 residual block (each one containing 2 convolutional layers and 1 rectified linear unit (ReLU)) are quite adequate to confirm the effectiveness of our strategy.

While a model's efficiency under DP can be improved by optimizing various factors, e.g., global training strategy [46], fine-tuning parameters [167], advanced optimizer [125] and model architecture [111], GeoDP is the first to focus on perturbation strategy, and makes no change to the logic of SGD so as to maintain its universal compatibility with any optimization that integrates SGD. As such, we believe basic models such as LR and CNN are more appropriate, as their interpretability can more accurately match the setting in our theoretical derivation. Such strategy has also

been adopted in a few other theoretical works, such as [80], which only utilizes LR to verify the convergence analysis.

5.3.1.2 Competitive Methods

As GeoDP is orthogonal to existing optimization techniques as interpreted in Section 2.2.3, we do not directly compare them. Instead, we compare GeoDP with DP on regular SGD from various perspectives, i.e., model efficiency, compatibility with existing optimization techniques. To demonstrate generality of GeoDP, we also apply a state-of-the-art clipping technique AUTO-S [17] to observe its improvements on GeoDP.

5.3.2 GeoDP vs. DP: Accuracy of Descent Trend

On the synthetic dataset, we perturb gradients by GeoDP and DP, respectively, and compare their MSEs under various parameters. As illustrated in Figure 5.3, labels θ and g represent MSEs of perturbed directions and gradients, respectively. In Figure 5.3(a)-5.3(c), we fix dimension $d = 5,000$ and batch size $B = 2,048$, while varying noise multiplier σ in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ if $\delta = 10^{-5}$) under three bounding factors $\beta = \{0.01, 0.1, 1\}$, respectively. We have two major observations. First, GeoDP better preserves directions (the red line is below the black line) while DP better preserves gradients (the blue line is below the green line) in most scenarios. Second, GeoDP is sometimes not robust to large noise multiplier and high dimensionality. When $\sigma > 1$ in Figure 5.3(a), GeoDP is instead outperformed by DP in preserving directions. Similar results can be also observed in Figure 5.3(d)-5.3(f) (fixing $\sigma = 8, B = 4096$ while varying dimensionality in $\{500, 1000, 2000, 5000, 10000, 20000\}$) and Figure 5.3(g)-5.3(i) (fixing $d = 10000, \sigma = 8$ while varying batch size in $\{512, 1024, 2048, 4096, 8192, 163984\}$), respectively. For example, Figure 5.3(d) and Figure 5.3(g), which all fix $\beta = 1$, show that GeoDP is

outperformed by DP on preserving directions when $d > 2000$ and $B < 8192$, respectively.

Before addressing this problem, we discuss reasons behind the ineffectiveness of GeoDP. Recall from Section 5.2.2 that the perturbation of GeoDP on directions is $\frac{\sqrt{d+2}\beta\pi}{B}\boldsymbol{n}_\sigma$. Obviously, both large noise multiplier ($\boldsymbol{n}_\sigma$) and high dimensionality ($\sqrt{d+2}$) increase the perturbation on directions.

Nevertheless, GeoDP can overcome this shortcoming by tuning β, which controls the sensitivity of direction. In both Figures 5.3(b) ($\beta = 0.1$) and 5.3(c) ($\beta = 0.01$), we reduce the noise on the direction by reducing the bounding factor, and the pay-off is very significant. Results show that GeoDP simultaneously outperforms DP in both direction and gradient. Tuning β is also effective in Figure 5.3(e), 5.3(f) and Figure 5.3(h), 5.3(i), respectively. Most likely, smaller bounding factor reduces noise added to the direction while does not affect the noisy magnitude. Accordingly, GeoDP reduces both MSEs of direction and gradient, and thus perfectly outperforms DP in preserving directional information.

To further confirm this conjecture, extensive experiments, by varying the bounding factor in $\{0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ under different scenarios, are conducted in Figure 5.4. All experimental results show that there always exists a bounding factor ($\beta = 0.2$ in Figure 5.4(a) and $\beta = 0.4$ in Figure 5.4(b) and $\beta = 0.8$ in Figure 5.4(c)) for GeoDP to outperform DP in preserving both direction and gradient. **These results also perfectly align with our theoretical analysis in Lemma 6 and Theorem 10, respectively.**

Also, GeoDP can improve accuracy by tuning batch size. As illustrated in Figure 5.3(g) ($d = 10000, \sigma = 8, \beta = 1$), we demonstrate how the performance of GeoDP is impacted by batch size. Obviously, a large batch size can boost GeoDP to provide optimal accuracy on directions. In contrast, the accuracy of DP on directions hardly changes with batch size (see the black line in 5.3(g)), although the noise scale on

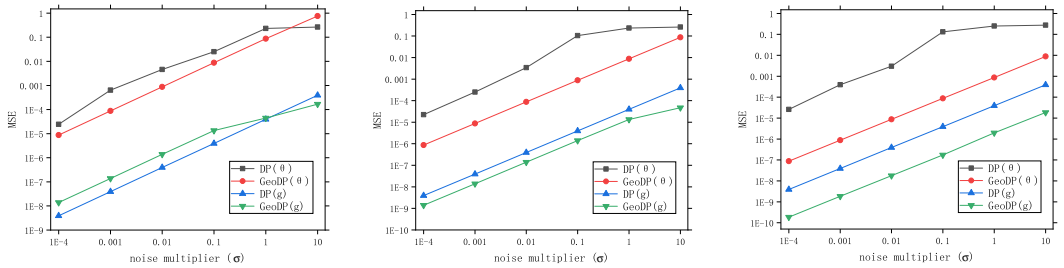
gradients is reduced by larger batch size (see the blue line in 5.3(g)). These results validate that **optimization techniques of DP-SGD**, such as fine-tuning learning rate, clipping threshold and batch size, **cannot reduce the noise on the direction, as confirmed by Corollary 2.**

5.3.3 GeoDP vs. DP: Logistic Regression

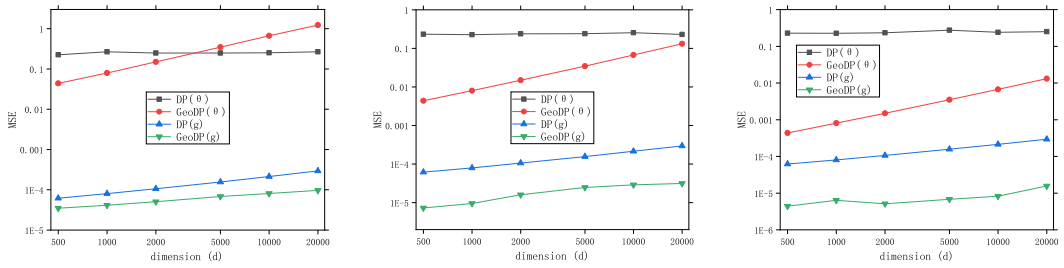
In the second set of experiments, we verify the effectiveness of GeoDP on Logistic Regression (LR) under MNIST dataset. Figure 5.5 plots training losses of 350 iterations, under *No noise*, *GeoDP* and *DP*. In Figure 5.5(a), with $B = 4,096$, GeoDP (the red line) significantly outperforms DP (the green line) and almost has the same performance as noise-free training (black line). The green line overlaps with the purple line because losses of DP-SGD with $B = 2,048$ and $B = 4,096$ are almost the same. This observation coincides with that from Figure 5.3(g), i.e., the batch size of DP-SGD hardly impacts the noise on the descent trend and thus the model efficiency. In contrast, batch size can successfully reduce the noise of GeoDP (see the gap between the red and blue lines).

In Figure 5.5(b), we test the performance of GeoDP under large noise scale. Initially, GeoDP (blue line) performs worse than DP (green line) with $\beta = 1$. When reducing β to 0.5 as suggested in Section 5.3.2, the performance of GeoDP surges and leaves DP behind. This observation confirms the superiority of GeoDP over DP even under extreme cases.

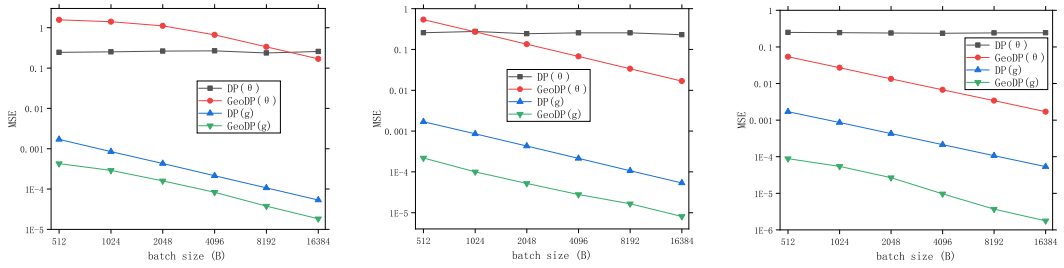
In Figure 5.5(c), we fix the $\beta = 1$ and $B = 256$ while varying the noise multiplier in $\sigma = \{0.01, 0.1\}$. As we can see, reducing σ cannot help DP to perform better (see the green line). This is because DP introduces biased noise to the direction, as confirmed by Lemma 6. Simply reducing the variance of noise cannot counteract this bias. As such, **DP is sub-optimal even under very small multiplier.** By contrast, GeoDP can achieve significant efficiency improvement with multiplier



(a) $d = 5000, B = 2048, \beta = 1$ (b) $d = 5000, B = 2048, \beta = 0.1$ (c) $d = 5000, B = 2048, \beta = 0.01$

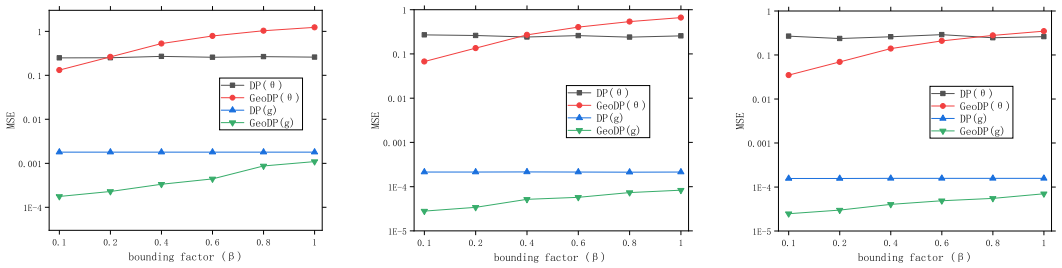


(d) $\sigma = 8, B = 4096, \beta = 1$ (e) $\sigma = 8, B = 4096, \beta = 0.1$ (f) $\sigma = 8, B = 4096, \beta = 0.01$



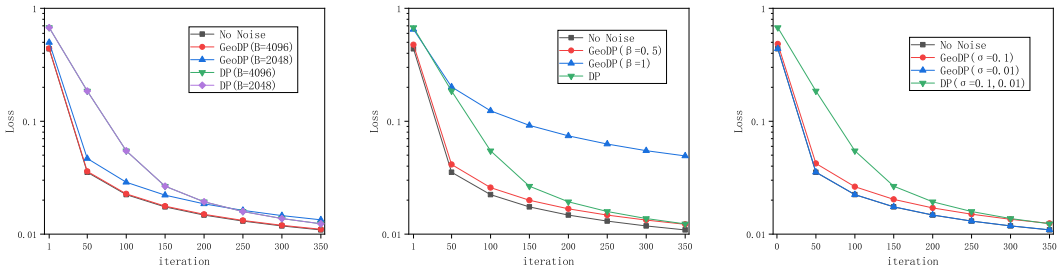
(g) $d = 10000, \sigma = 8, \beta = 1$ (h) $d = 10000, \sigma = 8, \beta = 0.1$ (i) $d = 10000, \sigma = 8, \beta = 0.01$

Figure 5.3: GeoDP vs. DP on Preserving Gradients under Various Parameters on Synthetic Dataset

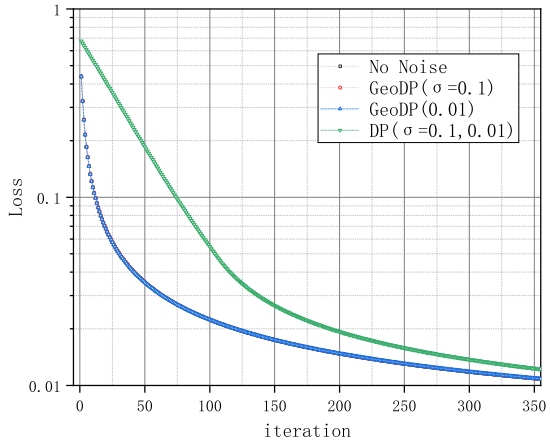


(a) $d = 20000, \sigma = 8, B = 4096$ (b) $d = 10000, \sigma = 8, B = 4096$ (c) $d = 5000, \sigma = 8, B = 4096$

Figure 5.4: The Effectiveness of Bounding Factor



(a) $d = 785, \sigma = 1, \beta = 1$ (b) $d = 785, \sigma = 10, B = 2,048$ (c) $d = 785, \beta = 1, B = 256$



(d) $d = 785, \beta = 1, B = 2,048$

Figure 5.5: GeoDP versus DP on Logistic Regression under MNIST dataset

reduction. When $\sigma = 0.01$ (see the blue line), GeoDP almost achieves noise-free model efficiency (the blue line is only slightly above the black line).

Similar results can be also observed in Figure 5.5(d), where GeoDP even achieves noise-free model efficiency with both $\sigma = 0.01, 0.1$ while DP cannot further achieve better model efficiency with noise multiplier reduced.

Dataset	Method	$\sigma = 10$	$\sigma = 1$
	DP ($B = 8192$)	87.93%	94.25%
	DP ($B = 16384$)	88.12%	95.52%
	DP($B = 16384$	88.40%	95.71%
MNIST	+AUTO-S)		
(noise-free	GeoDP ($B = 8192, \beta = 0.1$)	90.31%	96.47%
99.11%)	GeoDP ($B = 16384, \beta = 0.1$)	93.58%	98.04%
	GeoDP ($B = 8192, \beta = 0.5$)	53.80%	60.31%
	GeoDP ($B = 16384, \beta = 0.1$	93.64%	98.17%
	+AUTO-S)		

Table 5.2: GeoDP vs. DP on CNN under MNIST Dataset: Test Accuracy

5.3.4 GeoDP vs. DP: Deep Learning

To demonstrate the effectiveness of GeoDP in various learning tasks, we also conduct experiments on MNIST dataset with Convolutional Neural Network (CNN) and Residual . Due to the extremely large number of parameters, we set the number of training epochs to 20. While GeoDP pays much attention on the direction, the noisy magnitude is also impacting the overall model efficiency. This is why GeoDP also clips the magnitude before adding noise to it (see Step 6 in Algorithm 5). Since the L_2-norm of the gradient (i.e., the magnitude) is clipped in existing works [17, 164], the same techniques can also be applied to GeoDP. As such, we also demonstrate

the generality of GeoDP by integrating it to the state-of-the-art clipping technique AUTO-S [17].

Major results are demonstrated in Table 5.2. In general, GeoDP outperforms DP under various parameters except for large β. We can observe that the test accuracy is dramatically reduced (e.g., $98.7\% \rightarrow 60.3\%$) when β increases from 0.1 to 0.5. The reason behind is the extremely large sensitivity of GeoDP incurred by high dimensionality ($21,840$ dimensions), as discussed in 5.3.2. Overall, we can always find such a β ($\beta = 0.1$ in this experiment) that GeoDP outperforms DP in any task. Similar results in Table 5.3 also demonstrates the effectiveness of GeoDP on ResNet under CIFAR-10 dataset. Similar to our observations on LR, GeoDP even better outperforms DP under smaller noise multiplier (e.g., GeoDP can achieve better accuracy than DP even under $\beta = 1$.). **Note that the perturbed direction of GeoDP is unbiased while that of DP is biased, as previously confirmed in Lemma 6**. As such, the optimality of GeoDP over DP under smaller noise multiplier is a reflection of this nature.

Dataset	Method	$\sigma = 0.1$	$\sigma = 0.01$
CIFAR-10 (noise-free 67.43%)	DP ($B = 8192$)	59.39%	63.27%
	DP ($B = 16384$)	60.12%	63.84%
	DP($B = 16384$ +AUTO-S)	60.51%	63.91%
	GeoDP ($B = 8192, \beta = 1$)	61.47%	65.93%
	GeoDP ($B = 16384, \beta = 1$)	63.38%	66.51%
	GeoDP ($B = 16384, \beta = 0.1$)	65.47%	67.35%
	GeoDP ($B = 16384, \beta = 0.1$ +AUTO-S)	65.58%	67.37%

Table 5.3: GeoDP vs. DP on ResNet under CIFAR-10 Dataset: Test Accuracy

5.3.5 GeoDP versus DP: The Defense on MIA

As demonstrated earlier, GeoDP makes better trade-off than DP in SGD tasks. As such, it is worth investigating the integrity of GeoDP in face of MIA attacks. In particular, ML-Doctor [118] provides us a convenient platform to evaluate the ability of GeoDP and DP against MIA attacks. In particular, we implement white-box attack [95] of this framework with 70% random-chosen samples of the target training dataset as the auxiliary dataset. We follow the instruction [87] and feed four inputs to the attack model, i.e., the ranked posteriors of target samples, gradients from target model's last layer, classification loss and the true label. The model that is attacked is the two-layer CNN model above. With the batch size $B = 64$ and the total epoch 50, major results are summarized in Table 5.4. As we can see, GeoDP does not make MIA much more easier. Most likely, existing MIA attacks and GeoDP has different views on "similar gradients". For MIA, this similarity is numerical while for GeoDP, it is directional. As such, gradients from GeoDP can better deceive the attack model while GeoDP achieves satisfying model efficiency.

Method	$\epsilon = 4.9$	$\epsilon = 11.3$
Original Model(CelebA)	68.3%	68.3%
DP(B=8,192) (CelebA)	50.0%	50.0%
GeoDP(B=8,192,$\beta = 0.1$) (CelebA)	49.8%	50.0%
Original Model(FMNIST)	56.2%	56.2%
DP(B=8,192) (FMNIST)	50.0%	50.0%
GeoDP(B=8,192,$\beta = 0.1$) (FMNIST)	50.0%	50.1%

Table 5.4: GeoDP versus DP on ML-Doctor: Attack Accuracy

5.4 Summary

This chapter optimizes DP-SGD from a new perspective. We first theoretically analyze the impact of DP noise on the training process of SGD, which shows that the perturbation of DP-SGD is actually sub-optimal because it introduces biased noise to the direction. This inspires us to reduce the noise on direction for model efficiency improvement. We then propose our geometric perturbation mechanism GeoDP. Its effectiveness and generality are mutually confirmed by both rigorous proofs and experimental results. As for future work, we plan to study the impact of mainstream training optimizations, such as Adam optimizer [125], on GeoDP. Besides, we also plan to extend GeoDP to other form of learning, such as federated learning [47].

Chapter 6

Analyzing and Enhancing LDP Perturbation in Federated Learning

With the increasingly stringent legislation on personal data protection such as General Data Protection Regulation (GDPR) [102], federated learning (FL) [150], a decentralized machine learning paradigm to train a global model across multiple local devices, has become increasingly popular over traditional centralized machine learning. Besides their advantages in privacy preservation, most FL frameworks, commonly implementing prevalent deep learning algorithms (e.g., CNN) in local devices, are highly compatible with existing optimization techniques (i.e., mini-batch stochastic gradient descent (SGD) and its variants). These advantages have helped FL to embrace wider applications in practice. However, recent studies show that the training process in FL, especially the disclosure of local model weights or gradients, can still leak private information and is thus vulnerable to various privacy attacks, including membership inference attacks [91,93,118,157], attribute inference attacks [53,66], and data extraction attacks [20, 54]. These attacks pose immediate threats to the wider

adoption of FL in business sectors such as healthcare and finance where training data are sensitive.

To remedy this, local differential privacy (LDP) [32], which sends the perturbed data to any third party while reserving original information locally, is adopted in the industry [1, 124]. Referred to as federated LDP-SGD, random noise is added to the local gradient or other derived local parameters before sent to the central server. As such, federated LDP-SGD effectively prevents privacy attacks on model parameters, as true parameters always remain locally.

However, LDP-SGD is a strict privacy scheme and thus causes poor model efficiency. There are a few works that attempt to improve its performance but they all have limitations. First, among various LDP mechanisms, only Gaussian mechanism, which merely provides a relaxed privacy guarantee, is explored [50, 70, 86, 108, 114, 129, 141]. Second, LDP-SGD algorithm in prior works [70, 86] is not so effective as to provide satisfactory model efficiency even under Gaussian noise. Last but not the least, no prior works are able to generally evaluate the performances of different LDP mechanisms in federated training.

In this thesis, we address these limitations by first proposing an analytical framework that generalizes federated LDP-SGD and derives the impact of LDP noise on the federated training process, in terms of the model efficiency. Then we show that this framework can serve as a benchmark to compare model efficiencies of federated LDP-SGD under various LDP mechanisms. An interesting observation is that while existing works preserve the gradient itself, our analysis points out that **only its direction is necessary for gradient descent.** As such, existing LDP-SGD strategy is sub-optimal, as it wastes privacy budget to preserve the magnitude of gradient. Motivated by this, our second contribution is a geometric perturbation strategy *LDPVec* to optimize the training process. While focusing on preserving directional information, ***LDPVec* only perturbs the direction of a gradient, and rearranges LDP noise to better preserve directional information.** This strategy can generally

enhance federated LDP-SGD under various LDP mechanisms. To summarize, the main contributions of this chapter are as follows.

- To the best of our knowledge, this is the first general analytical framework to measure LDP mechanisms in federated learning. This framework can not only serve as a benchmark to compare various LDP mechanisms in federated LDP-SGD, but also point out the direction of future optimization.
- We propose a geometric perturbation strategy *LDPVec*, which optimizes performances of various LDP mechanisms in federated SGD.
- Extensive experiments on real datasets, popular machine learning models, and three state-of-the-art LDP mechanisms are conducted to validate the generality and effectiveness of both framework and strategy. All results unanimously show that the theoretical analysis is consistent with the experimental results, and our geometric perturbation strategy significantly improves model efficiencies in practice.

The rest of this chapter is organized as follows. Section 6.1 presents the analytical framework for federated LDP-SGD while Section 6.2 proposes the perturbation strategy *LDPVec*. Experimental results are presented in Section 6.3, and summaries are drawn in Section 6.4.

6.1 A General Analytical Framework for Federated LDP-SGD

In this section, we propose an analytical framework to generally analyze the model efficiency of federated LDP-SGD. In what follows, Section 6.1.1 generalizes federated LDP-SGD tasks, based on which we further model the global aggregation in Section

6.1.2, while conducting model efficiency analysis in Section 6.1.3. Finally, Section 6.1.4 presents a case study on federated logistic regression under Laplace, Piecewise and Gaussian mechanisms, respectively, to demonstrate the implementation of our framework.

6.1.1 Overview of Federated LDP-SGD

Let us assume there are T iterations in total, and one global aggregation occurs per E iterations. As such, there are T/E rounds of federated LDP-SGD. As shown in Figure 6.1, each round of federated LDP-SGD has five stages and finally terminates at stage ⑥.

- *Local Regular Iterations:* On receiving the current global model $\boldsymbol{w}^{t*}$, each local device first updates its local model $\boldsymbol{w}_k^t = \boldsymbol{w}^{t*}$, and then initiates $E-1$ times local SGD as $\boldsymbol{w}_k^{t+i} = \boldsymbol{w}_k^{t+i-1} - \eta^{t+i-1}\tilde{\boldsymbol{g}}_k^{t+i-1}, i = 1, 2, ..., E-1$.

- *Local Noisy Iteration:* On the E-th iteration, the gradient of the E-th local SGD is added by one random LDP noise. Formally,

$$\begin{aligned} \boldsymbol{w}_k^{t+E*} &= \boldsymbol{w}_k^{t+E-1} - \eta^{t+E-1}\boldsymbol{g}_k^{t+E-1*} \\ &= \boldsymbol{w}_k^{t+E-1} - \eta^{t+E-1}(\tilde{\boldsymbol{g}}_k^{t+E-1} + \boldsymbol{n}_k^{t+E-1}) \end{aligned} \tag{6.1}$$

- *To Server:* The perturbed models $\left\{\boldsymbol{w}_k^{t+E*} | 1 \leq k \leq N\right\}$ are sent to the central server while the original models remain locally.

- *Global Aggregation:* On receiving perturbed models from all devices, the central server aggregates perturbed models to derive the current global model. Formally,

$$\boldsymbol{w}^{t+E*} = \sum_{k=1}^{N} p_k \boldsymbol{w}_k^{t+E*} \tag{6.2}$$

- *Global Broadcast:* The central server updates the global model to each local device for the next round of local iterations. Namely, $\boldsymbol{w}_k^{t+E} = \boldsymbol{w}^{t+E*}$.

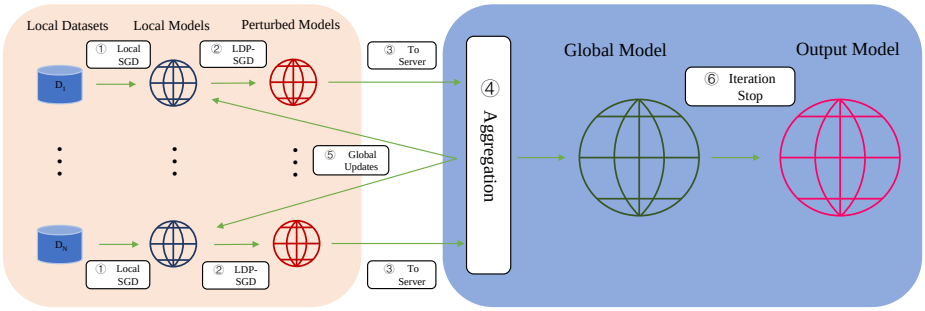


Figure 6.1: Overview of Federated LDP-SGD.

- *Global Stop:* LDP-SGD terminates when a satisfactory global model is derived.

In a complete round of federated LDP-SGD (Stages ①-⑤), Stages ①-② are "the local iterations" and Stages ③-⑤ are "the global update", respectively. Obviously, the LDP noise added on the local iteration (i.e., Stage ②) directly impacts the convergence on the global aggregation (i.e., Stage ④). The current convergence state is then passed on to the following local iterations via the global broadcast (i.e., Stage ⑤), which further feeds back to the future convergence. To provide a both accurate and user-friendly benchmark, we just analyze how LDP noise impacts one complete round of training, to compare performances of various LDP mechanisms.

6.1.2 Model of Federated LDP-SGD

As Stages ③,⑤ and ⑥ are already described in Section 6.1.1, we turn our focus Stages ①,② and ④, namely, local iterations and the global aggregation. Let Γ_E denote the set of global updates, i.e., $\Gamma_E = \{nE | n = 1, 2, ..., T/E\}$. When $t+1 \in \Gamma_E$, it is the time to perform the global update after one noisy local iteration. Otherwise, it is just a local iteration. For simplicity, we can generalize local iterations (i.e., Stages ① and ②) in the k-th device as:

$$\boldsymbol{w}_k^{t+1*} = \begin{cases} \boldsymbol{w}_k^t - \eta^t \tilde{\boldsymbol{g}}_k^t, & \text{if} \quad t+1 \notin \Gamma_E \\ \boldsymbol{w}_k^t - \eta^t (\tilde{\boldsymbol{g}}_k^t + \boldsymbol{n}_k^t), & \text{if} \quad t+1 \in \Gamma_E. \end{cases} \tag{6.3}$$

Following Equation 6.3, the global aggregation at Stage ④ (i.e., $t+1 \in \Gamma_E$) is modeled as:

$$\boldsymbol{w}^{t+1*} = \sum_{k=1}^{N} p_k \boldsymbol{w}_k^{t+1*}. \tag{6.4}$$

Applying Equation 6.3 to Equation 6.4, we have :

$$\begin{aligned} \boldsymbol{w}^{t+1*} &= \sum_{k=1}^{N} p_k \boldsymbol{w}_k^{t+1*} = \sum_{k=1}^{N} p_k (\boldsymbol{w}_k^t - \eta^t (\tilde{\boldsymbol{g}}_k^t + \boldsymbol{n}_k^t)) \\ &= \sum_{k=1}^{N} p_k (\boldsymbol{w}_k^t - \eta^t \tilde{\boldsymbol{g}}_k^t) - \eta^t \underbrace{\sum_{k=1}^{N} p_k \boldsymbol{n}_k^t}_{A}. \end{aligned} \tag{6.5}$$

As indicated in Equation 6.5, in local iterations two essential terms, namely the gradient itself and the LDP noise, jointly determine the global aggregation at Stage ④. While the gradient can be derived from the loss function, the random noise itself cannot be modeled. On the other hand, the aggregation of LDP noise (Item A) seems to be described by *Lindeberg–Lévy Central Limit Theorem* (CLT) [30]. However, since different local devices in FL may have different weights, i.e., $\{p_k | 1 \leq k \leq N\}$, the sampled noise $p_k \boldsymbol{n}_k^t$ is non-identically distributed. This violates the prerequisite of CLT. Fortunately, we establish the following lemma, which uses a virtual aggregation $\bar{p} \sum_{k=1}^{N} \boldsymbol{n}_k^t$ to replace Item A, where $\bar{p} = \frac{\sum_{k=1}^{N} p_k}{N}$.

Lemma 8. *Item A has the same distribution as* $\bar{p} \sum_{k=1}^{N} \boldsymbol{n}_k^t$.

Proof. Given $\tilde{\boldsymbol{g}}_k^t$, $\boldsymbol{g}_k^{t*}$ is only decided by the LDP mechanism and the privacy budget. Therefore, $\boldsymbol{n}_k^t = \boldsymbol{g}_k^{t*} - \tilde{\boldsymbol{g}}_k^t$ follows the same distribution. Suppose the *pdf* function of $\boldsymbol{n}_k^t$ is $f(\boldsymbol{x})$. Then, we have:

$$\begin{aligned} &\sum_{k=1}^{N} p_k \boldsymbol{n}_k^t \int_{-\infty}^{\boldsymbol{x}} f(\boldsymbol{n}_k^t) d\boldsymbol{n}_k^t = \sum_{k=1}^{N} p_k \boldsymbol{n} \int_{-\infty}^{\boldsymbol{x}} f(\boldsymbol{n}) d\boldsymbol{n} \\ =&\bar{p} N \boldsymbol{n} \int_{-\infty}^{\boldsymbol{x}} f(\boldsymbol{n}) d\boldsymbol{n} = \bar{p} \sum_{k=1}^{N} \boldsymbol{n}_k^t \int_{-\infty}^{\boldsymbol{x}} f(\boldsymbol{n}_k^t) d\boldsymbol{n}_k^t. \end{aligned} \tag{6.6}$$

by which our claim is proven. □

Since $\boldsymbol{n}_k^t$ is identically distributed, CLT [30] can describe the above virtual aggregation, namely, $\bar{p}\sum_{k=1}^{N}\boldsymbol{n}_k^t$. The same model also describes Item A, as proven by Lemma 8. As we respectively assume that $\boldsymbol{\mu}_\nu^t : \{\boldsymbol{\mu}_{\nu z}^t = \mathbb{E}\left(\boldsymbol{n}_{\nu kz}^t\right) | 1 \le z \le d\}$ and $\boldsymbol{\sigma}_\nu^t : \left\{{\boldsymbol{\sigma}_{\nu z}^t}^2 = \mathbb{E}\left({\boldsymbol{n}_{\nu kz}^t}^2\right) - \mathbb{E}^2\left(\boldsymbol{n}_{\nu kz}^t\right) | 1 \le z \le d\right\}$, the following theorem derives Item A with a Gaussian distribution. However, no existing study instructs the strict LDP mechanisms to solve $\boldsymbol{n}_k^t$ given the clipped gradient $\tilde{\boldsymbol{g}}_k^t$. Following the same approach as existing LDP-SGD works [70,86,141], given the clipped gradient $\tilde{\boldsymbol{g}}_k^t(\left\|\tilde{\boldsymbol{g}}_k^t\right\| \le C)$, its normalized gradient is $\tilde{\boldsymbol{g}}_k'^t = \tilde{\boldsymbol{g}}_k^t\sqrt{d}/C(\left\|\tilde{\boldsymbol{g}}_k'^t\right\| \le \sqrt{d})$. Referring to Section 3.1.1, we can produce the normalized LDP noise $\boldsymbol{n}_k'^t$. While the normalized noise $\boldsymbol{n}_k'^t$ guarantees that the normalized perturbed gradient $\boldsymbol{g}_k'^{t*} = \tilde{\boldsymbol{g}}_k'^t + \boldsymbol{n}_k'^t$ follows LDP, $\boldsymbol{n}_{kz}^t = \boldsymbol{n}_{kz}'^t C/\sqrt{d}$ guarantees that the perturbed gradient $\boldsymbol{g}_{kz}^{t*} = \tilde{\boldsymbol{g}}_{kz}^t + \boldsymbol{n}_{kz}^t$ also follows LDP, as guaranteed by the following lemma.

Lemma 9. *Let $\mathcal{M} : D \to R$ be a (ϵ, δ)-locally differentially private mechanism, and $f : J \to D$ be a linear mapping. Then $\mathcal{M}(f) : J \to R$ is also (ϵ, δ)-locally differentially private.*

Proof. For any pair of original data $\boldsymbol{g}_i', \boldsymbol{g}_j' \in D$, any perturbed data $\boldsymbol{g}'^* \in R$ and their respective linear mapping $\boldsymbol{g}_i = f(\boldsymbol{g}_i')$, $\boldsymbol{g}_j = f(\boldsymbol{g}_j')$, $\boldsymbol{g}^* = f(\boldsymbol{g}'^*)$, we have:

$$\begin{aligned} &\Pr\left(f(\mathcal{M}(\boldsymbol{g}_i')) = \boldsymbol{g}^*\right) = \Pr\left(\mathcal{M}(\boldsymbol{g}_i') = \boldsymbol{g}'^*\right) \\ &\le \exp(\epsilon)\Pr\left(\mathcal{M}(\boldsymbol{g}_j')\right) + \delta = \exp(\epsilon)\Pr\left(f(\mathcal{M}(\boldsymbol{g}_j')) = \boldsymbol{g}^*\right) + \delta. \end{aligned} \tag{6.7}$$

Then we prove the equivalency of $f(\mathcal{M}(\boldsymbol{g}_i'))$ and $\mathcal{M}(\boldsymbol{g}_i)$. Suppose $f(\boldsymbol{x}) = c\boldsymbol{x}$, we have:

$$f(\mathcal{M}(\boldsymbol{g}_i')) = f(\boldsymbol{g}_i' + \boldsymbol{n}_i') = c\boldsymbol{g}_i' + c\boldsymbol{n}_i' = \boldsymbol{g}_i + \boldsymbol{n}_i = \mathcal{M}(\boldsymbol{g}_i). \tag{6.8}$$

Applying Equation 6.8 to Equation 6.7, we have $\Pr\left(\mathcal{M}(\boldsymbol{g}_i) = \boldsymbol{g}^*\right) \le \exp(\epsilon)\Pr\left(\mathcal{M}(\boldsymbol{g}_j) = \boldsymbol{g}^*\right) + \delta$, by which this lemma is proven. □

Theorem 12. *The distribution of Item A is $\mathcal{N}(C\bar{p}_k N\boldsymbol{\mu}_\nu^t, C^2\bar{p}_k^2 N(\boldsymbol{\sigma}_\nu^t)^2)$.*

Proof. For $\forall k$, $\boldsymbol{n}_k'^t$ are independent and identically distributed (i.i.d.) random variables. As per *Lindeberg–Lévy Central Limit Theorem* [30,43,115], the following probability always holds if $N \to \infty$:

$$\begin{aligned}
&\Pr\left(\frac{\sum_{k=1}^{N} \boldsymbol{n}_{\nu kz}^t - N\boldsymbol{\mu}_{\nu z}^t}{\sqrt{N}\boldsymbol{\sigma}_{\nu z}^t} \leq X\right) = \Pr\left(\frac{\frac{1}{C}\sum_{k=1}^{N} \boldsymbol{n}_{kz}^t - N\boldsymbol{\mu}_{\nu z}^t}{\sqrt{N}\boldsymbol{\sigma}_{\nu z}^t} \leq X\right) \\
&= \Pr\left(\frac{\sum_{k=1}^{N} \boldsymbol{n}_{kz}^t - CN\boldsymbol{\mu}_{\nu z}^t}{\sqrt{N}C\boldsymbol{\sigma}_{\nu z}^t} \leq X\right) \\
&= \Pr\left(\frac{\bar{p}_k\sum_{k=1}^{N} \boldsymbol{n}_{kz}^t - C\bar{p}_k N\boldsymbol{\mu}_{\nu z}^t}{\sqrt{N}\bar{p}_k C\boldsymbol{\sigma}_{\nu z}^t} \leq X\right) = \int_{-\infty}^{X} \phi(x)dx.
\end{aligned} \tag{6.9}$$

where $\phi(x)$ is the *pdf* of the standard Gaussian distribution $\mathcal{N}(0,1)$. Therefore, $\lim_{N\to\infty} \frac{\bar{p}_k\sum_{k=1}^{N} \boldsymbol{n}_{kz}^t - C\bar{p}_k N\boldsymbol{\mu}_{\nu z}^t}{\sqrt{N}C\bar{p}_k\boldsymbol{\sigma}_{\nu z}^t}$ follows standard normal distribution. Then we derive the distribution of $\lim_{N\to\infty} \bar{p}_k\sum_{k=1}^{N} \boldsymbol{n}_{kz}^t$ as

$$\lim_{N\to\infty} \bar{p}_k \sum_{k=1}^{N} \boldsymbol{n}_{kz}^t \sim \mathcal{N}(\frac{C\bar{p}_k N}{\sqrt{d}}\boldsymbol{\mu}_z^t, \frac{C^2\bar{p}_k^2 N}{d}{\boldsymbol{\sigma}_z^t}^2). \tag{6.10}$$

Due to the independence of each dimension of the clipped gradient, we can further derive the distribution of $\lim_{N\to\infty} \bar{p}_k\sum_{k=1}^{N} \boldsymbol{n}_k^t$:

$$\lim_{N\to\infty} \bar{p}_k \sum_{k=1}^{N} \boldsymbol{n}_k^t \sim \mathcal{N}(C\bar{p}_k N\boldsymbol{\mu}_\nu^t, C^2\bar{p}_k^2 N{\boldsymbol{\sigma}_\nu^t}^2). \tag{6.11}$$

As proven by Lemma 8, item A shares the same distribution as $\bar{p}_k \lim_{N\to\infty} \sum_{k=1}^{N} \boldsymbol{n}_k^t$, by which this theorem is proven. □

From Equation 6.9, gradient clipping, which reduces both the expectation and the variance of LDP noise, effectively reduce the magnitude of the LDP noise. However, it remains unclear how LDP noise impacts on the convergence. Now that we have mathematically derived a complete round of federated LDP-SGD, it is ready to analyze its convergence.

6.1.3 A General Analytical Framework for Federated LDP-SGD

Let a virtual[1] global gradient $\tilde{\boldsymbol{g}}^t = \sum_{k=1}^N p_k \tilde{\boldsymbol{g}}_k^t = \sum_{k=1}^N p_k \nabla F_k(\boldsymbol{w}_k^t, S_k^t)$, a virtual global noise $\boldsymbol{n}^t = \sum_{k=1}^N p_k \boldsymbol{n}_k^t$, and a virtual global model $\boldsymbol{w}^t = \sum_{k=1}^N p_k \boldsymbol{w}_k^t$. Then a virtual global SGD of federated LDP-SGD $\boldsymbol{w}^{t+1*} = \sum_{k=1}^N p_k \boldsymbol{w}_k^t - \eta^t \sum_{k=1}^N p_k \tilde{\boldsymbol{g}}_k^{t*} = \boldsymbol{w}^t - \eta^t \left(\tilde{\boldsymbol{g}}^t + \boldsymbol{n}^t\right)$, and a virtual global model of federated SGD $\boldsymbol{w}^{t+1} = \sum_{k=1}^N p_k \boldsymbol{w}_k^t - \eta^t \sum_{k=1}^N p_k \tilde{\boldsymbol{g}}_k^t = \boldsymbol{w}^t - \eta^t \tilde{\boldsymbol{g}}^t$. Since $\nabla_k^t = \mathbb{E}(\tilde{\boldsymbol{g}}_k^t)$, we also have a virtual expectation of a global gradient $\nabla^t = \mathbb{E}(\tilde{\boldsymbol{g}}^t) = \sum_{k=1}^N p_k \nabla_k^t$. In general, the Euclidean distances between global models and the global optima (i.e., $\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\|^2$ and $\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\|^2$) reflect the model efficiency of federated LDP-SGD and federated SGD, respectively. Apparently, the smaller this distance is, the better efficiency the global model achieves. Their efficiency difference (i.e., $\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\|^2$), on the other hand, can describe the impact of LDP noise on the global model, as presented by the following theorem.

Theorem 13. *Given $t+1 \in \Gamma_E$, the impact of LDP noise on federated LDP-SGD can be measured as:*

$$\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\right\|^2\right) - \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\right\|^2\right) \\
&= -2\underbrace{\left\langle \boldsymbol{w}^t - \eta^t \nabla^t, C\bar{p}_k N \boldsymbol{\mu}_\nu^t \right\rangle}_{B} \\
&+ \underbrace{C^2 {\eta^t}^2 \bar{p}_k^2 N^2 (\boldsymbol{\mu}_\nu^t)^2 + C^2 {\eta^t}^2 \bar{p}_k^2 N (\boldsymbol{\sigma}_\nu^t)^2}_{C} + 2\underbrace{\left\langle \boldsymbol{w}^\star, C\bar{p}_k N \boldsymbol{\mu}_\nu^t \right\rangle}_{D}.
\end{aligned} \tag{6.12}$$

Proof. From federated LDP-SGD, we have:

$$\begin{aligned}
&\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\right\|^2 = \left\|\boldsymbol{w}^t - \eta^t \left(\tilde{\boldsymbol{g}}^t + \boldsymbol{n}^t\right) - \boldsymbol{w}^\star\right\|^2 \\
&= \left\|\boldsymbol{w}^t - \eta^t \tilde{\boldsymbol{g}}^t - \boldsymbol{w}^\star\right\|^2 - 2\langle \boldsymbol{w}^t - \eta^t \tilde{\boldsymbol{g}}^t - \boldsymbol{w}^\star, \boldsymbol{n}^t \rangle + {\eta^t}^2 {\boldsymbol{n}^t}^2.
\end{aligned} \tag{6.13}$$

[1]By "virtual", we mean that they do not physically exist in the FL training process but are much convenient in terms of convergence analysis.

While for federated SGD, we have:

$$\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2 = \left\|\boldsymbol{w}^{t} - \eta^t \tilde{\boldsymbol{g}}^t - \boldsymbol{w}^{\star}\right\|^2. \tag{6.14}$$

Subtracting Equation 6.14 from Equation 6.13, we have:

$$\begin{aligned} &\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^{\star}\right\|^2 - \left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2 \\ =& -2\langle \boldsymbol{w}^t - \eta^t \tilde{\boldsymbol{g}}^t - \boldsymbol{w}^{\star}, \boldsymbol{n}^t\rangle + {\eta^t}^2 {\boldsymbol{n}^t}^2 \\ =& -2\underbrace{\langle \boldsymbol{w}^t - \eta^t \tilde{\boldsymbol{g}}^t, \boldsymbol{n}^t\rangle}_{B} + \underbrace{{\eta^t}^2 {\boldsymbol{n}^t}^2}_{C} + 2\underbrace{\langle \boldsymbol{w}^{\star}, \boldsymbol{n}^t\rangle}_{D}. \end{aligned} \tag{6.15}$$

Recall that from Equation 6.11, $\boldsymbol{n}^t$ approximates a Gaussian distribution. That is to say, $\boldsymbol{n}^t = \mathcal{N}(C\bar{p}_k N \boldsymbol{\mu}_\nu^t, C^2 \bar{p}_k^2 N {\boldsymbol{\sigma}_\nu^t}^2)$. Given a variable x, its expectation μ and its variance σ^2, the expectation of item B is:

$$\mathbb{E}(B) = \langle \boldsymbol{w}^t - \eta^t \mathbb{E}(\tilde{\boldsymbol{g}}^t), \mathbb{E}(\boldsymbol{n}^t)\rangle = \left\langle \boldsymbol{w}^t - \eta^t \nabla^t, C\bar{p}_k N \boldsymbol{\mu}_\nu^t \right\rangle. \tag{6.16}$$

Given a variable x, its expectation μ and its variance σ^2, since $\mathbb{E}(x-\mu)^2 = \mathbb{E}(x^2) - 2\mu\mathbb{E}(x) + \mu^2 = \mathbb{E}(x^2) - \mu^2 = \sigma^2$, we have $\mathbb{E}(x^2) = \mu^2 + \sigma^2$. On this basis, we derive the expectation of Item C:

$$\mathbb{E}(C) = {\eta^t}^2 \mathbb{E}({\boldsymbol{n}^t}^2) = C^2 {\eta^t}^2 \bar{p}_k^2 N^2 {\boldsymbol{\mu}_\nu^t}^2 + C^2 {\eta^t}^2 \bar{p}_k^2 N {\boldsymbol{\sigma}_\nu^t}^2. \tag{6.17}$$

For Item D, we have:

$$\mathbb{E}(D) = \langle \boldsymbol{w}^{\star}, \mathbb{E}(\boldsymbol{n}^t)\rangle = \left\langle \boldsymbol{w}^{\star}, C\bar{p}_k N \boldsymbol{\mu}_\nu^t \right\rangle. \tag{6.18}$$

Applying Equation 6.16, 6.17 and 6.18 into Equation 6.15, we prove this theorem. □

Overall, a positive difference, i.e., $\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^{\star}\right\|^2\right) > \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2\right)$, means that the efficiency of federated LDP-SGD is worse than federated SGD. This perception is highly recognized by most existing works [50, 70, 86, 141]. Zero difference, i.e., $\boldsymbol{\mu}_\nu^t = \boldsymbol{\sigma}_\nu^t = 0$, means no LDP perturbation. The most interesting scenario is when the difference is negative, i.e., $\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^{\star}\right\|^2\right) < \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2\right)$.

From Equation 6.15,the efficiency difference is divided into three parts. Item B, the inner product between the noise-free global model and the noise itself, reflects how the direction of LDP noise impacts the efficiency. Item C otherwise describes how the magnitude of noise impacts the efficiency. While item C is mostly non-negative, item B depends on the angle between the global gradient descent $\boldsymbol{w}^t - \eta^t \nabla^t$ and the global aggregation of noise $\boldsymbol{\mu}_\nu^t$. Item D, however, shows that the introduction of LDP noise would definitely cause a bias to the global optima. In fact, the following corollary indicates that **federated LDP-SGD deviates from the global optima in a statistic sense.**

Corollary 3. *The impact of LDP noise on federated LDP-SGD is convergent as follows:*

$$\begin{aligned} &\lim_{t\to\infty} \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^\star\right\|^2\right) - \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^\star\right\|^2\right) \\ =&C^2 {\eta^t}^2 \bar{p}_k^2 N^2 {\boldsymbol{\mu}_\nu^t}^2 + C^2 {\eta^t}^2 \bar{p}_k^2 N {\boldsymbol{\sigma}_\nu^t}^2. \end{aligned} \tag{6.19}$$

Proof. After sufficient iterations, federated SGD would finally converge to the global optima. As $\boldsymbol{w}^t - \eta^t \nabla^t$ models the non-LDP federated SGD, we have $\lim_{t\to\infty} \boldsymbol{w}^t - \eta^t \nabla^t \to \boldsymbol{w}^\star$. Applying this to Equation 6.12, we have Item B and Item D mutually canceled out. As such, this corollary is proven. □

Since this difference has a direct impact on the model efficiency, it can benchmark the utilities of various LDP mechanisms in federated LDP-SGD. In particular, $\boldsymbol{w}^t - \eta^t \nabla^t$ and $\boldsymbol{w}^\star$ can be derived according to the FL algorithms (e.g., FedAvg) and local datasets, while $\boldsymbol{\mu}_\nu^t$ and $\boldsymbol{\sigma}_\nu^t$ can be calculated from the LDP mechanisms. Although $\boldsymbol{w}^\star$ is unknown in practice, especially when the loss function is non-convex, it is a constant throughout the above derivation. Accordingly, we can calculate respective efficiency differences in terms of respective LDP mechanism under FedAvg. The more negative the distance is, the better performance this mechanism achieves.

6.1.4 A Case Study: Benchmarking LDP Mechanisms in Federated Logistic Regression

To demonstrate our analytical framework, in this subsection we provide a case study on federated logistic regression [150], where we benchmark the efficiency of federated logistic regression under three state-of-the-art LDP mechanisms, namely, Laplace, Piecewise and Gaussian mechanisms, in the global aggregation step, i.e., $t+1 \in \Gamma_E$. We assume an MNIST [76] dataset (60,000 28×28 images) distributed among $N = 500$ devices in a non-iid fashion where each device contains 120 images of only two digits.

For simplicity, we consider a binary classification task, where digits $0-4$ are labeled as "0", and digits $5-9$ are labeled as "1". Since each image has 784 pixels, the local model weight should have 785 dimensions (784 features plus 1 bias). Let $\boldsymbol{s}_{kj}$: $\left[1, \boldsymbol{s}_{kj}^1, \boldsymbol{s}_{kj}^2, ..., \boldsymbol{s}_{kj}^{783}, \boldsymbol{s}_{kj}^{784}\right]$ and y_{kj} respectively denote one image and its label, and $\boldsymbol{w}_k^t$: $[\boldsymbol{w}_{k0}^t, \boldsymbol{w}_{k1}^t, ..., \boldsymbol{w}_{k783}^t, \boldsymbol{w}_{k784}^t]$ denote the local model. Then we have a linear combination $v_{kj} = {\boldsymbol{w}_k^t}^\top \boldsymbol{s}_{kj} = \sum_{z=1}^{785} \boldsymbol{w}_{kz}^t \boldsymbol{s}_{kj}^z$. Given a sigmoid function $f(v) = \frac{1}{1+e^{-v}}$, we derive the probability of $\boldsymbol{s}_{kj}$ being identified as "0" and "1", respectively:

$$\Pr(y_{kj} = 1|\boldsymbol{w}; \boldsymbol{s}) = f(v_{kj}), \quad \Pr(y_{kj} = 0|\boldsymbol{w}; \boldsymbol{s}) = 1 - f(v_{kj}). \tag{6.20}$$

Since there are 120 image in each device, we define the maximum likelihood function in terms of $\boldsymbol{s}_{kj}$:

$$\begin{aligned} L(\boldsymbol{w}; \boldsymbol{s}_{kj}) &= \ln \Pr(y_{kj}|\boldsymbol{w}; \boldsymbol{s}_{kj}) \\ &= y_{kj} \ln f(v_{kj}) + (1 - y_{kj}) \ln(1 - f(v_{kj})). \end{aligned} \tag{6.21}$$

Obviously, the larger the likelihood, the more accurate the local model is. To derive the loss function $l(\boldsymbol{w}; \boldsymbol{s}_{kj}) = -L(\boldsymbol{w}; \boldsymbol{s}_{kj})$, the local objective function of logistic regression is defined as follows:

$$\begin{aligned} F_k(\boldsymbol{w}; S_k) &= \frac{1}{B} \sum_{j=1}^{B} l(\boldsymbol{w}; s_{kj}) \\ &= -\frac{1}{B} \sum_{j=1}^{B} \left(y_{kj} \ln f(v_{kj}) + (1 - y_{kj}) \ln(1 - f(v_{kj}))\right). \end{aligned} \tag{6.22}$$

where $B \leq 120$. For simplicity, we let $p_k = 1/N = 0.002$. Given $f'(v) = f(v)(1 - f(v))$, the gradient $\boldsymbol{g}_k^t$ of the t-th local iteration is:

$$\begin{aligned}
\boldsymbol{g}_k^t &= \nabla F_k(\boldsymbol{w}_k^t; S_k) \\
&= -\frac{1}{B}\sum_{j=1}^{B}\left(\frac{y_{kj}}{f(v_{kj})} - \frac{1-y_{kj}}{1-f(v_{kj})}\right) f'(v_{kj})\frac{\partial v}{\partial \boldsymbol{w}} \\
&= -\frac{1}{B}\sum_{j=1}^{B}\left(y_{kj}(1-f(v_{kj})) - (1-y_{kj})f(v_{kj})\right)\boldsymbol{s}_{kj} \\
&= -\frac{1}{B}\sum_{j=1}^{B}\left(y_{kj} - f(v_{kj})\right)\boldsymbol{s}_{kj} \\
&= -\frac{1}{B}\sum_{j=1}^{B}\left(y_{kj} - f({\boldsymbol{w}_k^t}^\top \boldsymbol{s}_{kj})\right)\boldsymbol{s}_{kj}.
\end{aligned} \tag{6.23}$$

For simplicity, let $\boldsymbol{w}_k^t = \boldsymbol{w}^t = \mathbf{0}$. By setting $C = 1.8$ and applying local datasets to Equation 6.23, we can derive:

$$\begin{aligned}
\nabla^t &= \sum_{k=1}^{N} p_k \nabla_k^t = \mathbb{E}(\tilde{\boldsymbol{g}}^t) = \mathbb{E}(\boldsymbol{g}^t)/\max\left\{1, \left\|\boldsymbol{g}_k^t\right\|/C\right\} \\
&= -\frac{1}{500\times 120}\sum_{k=1}^{500}\sum_{j=1}^{120}\frac{\left(y_{kj} - f({\boldsymbol{w}_k^t}^\top \boldsymbol{s}_{kj})\right)\boldsymbol{s}_{kj}}{\max\left\{1, \left\|\boldsymbol{g}_k^t\right\|/C\right\}} \\
&= \frac{1}{60000}\sum_{k=1}^{500}\sum_{j=1}^{120}\left(\frac{1}{2} - y_{kj}\right)\boldsymbol{s}_{kj}.
\end{aligned} \tag{6.24}$$

By setting $\eta^t = 0.1$, we can derive $\boldsymbol{w}^t - 0.1\nabla^t$. [2] Let $\boldsymbol{w}^\star$ denote a well-trained model without LDP noise. In what follows, we demonstrate how to compare the performances of federated logistic regression under different LDP mechanisms, while allocating privacy budget $\epsilon = 0.1$ to each dimension.

Laplace mechanism. First, we derive the expectation and the variance of the normalized noise $\boldsymbol{n}_{\nu kz}^t$. Namely,

$$\boldsymbol{\mu}_z^t = \mathbb{E}\left(\boldsymbol{n}_{\nu kz}^t\right) = 0,\ \boldsymbol{\sigma}_z^{t\,2} = \mathbb{E}\left({\boldsymbol{n}_{\nu kz}^t}^2\right) - \mathbb{E}^2\left(\boldsymbol{n}_{\nu kz}^t\right) = \frac{4}{\epsilon^2} = 400. \tag{6.25}$$

[2] We do not display ∇^t and $\boldsymbol{w}^t - 0.1\nabla^t$ because each has 785 dimensions.

Note that clipping threshold, measured with L_2-norm, cannot directly considered as the sensitivity of Laplace. In each dimension of gradients, the maximum change is $2C = 0.2$. Therefore, we calibrate the sensitivity as $C' = 785 * 2C = 157$. By applying Equation 6.25 to Equation 6.12, we have:

$$\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^{\star}\right\|^2\right) - \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2\right) \\
=& -2\left\langle \boldsymbol{w}^t - \eta^t \nabla^t, \mathbf{0}\right\rangle + C'^2 \eta^{t\,2} \bar{p}_k^2 N^2 \sum_{z=1}^{785} {\boldsymbol{\mu}_z^t}^2 \\
&+ C'^2 \eta^{t\,2} \bar{p}_k^2 N \sum_{z=1}^{785} {\boldsymbol{\sigma}_z^t}^2 + 2\left\langle \boldsymbol{w}^{\star}, \mathbf{0}\right\rangle \approx 25,521,315.
\end{aligned} \tag{6.26}$$

Piecewise mechanism. According to [30], we have:

$$\begin{aligned}
\boldsymbol{\mu}_z^t &= \mathbb{E}\left(\boldsymbol{n}_{\nu kz}^t\right) = 0, \\
{\boldsymbol{\sigma}_z^t}^2 &= \mathbb{E}\left({\boldsymbol{n}_{\nu kz}^t}^2\right) - \mathbb{E}^2\left(\boldsymbol{n}_{\nu kz}^t\right) \\
&= \frac{{\nabla_z^t}^2}{2\exp(\epsilon/2) - 2} + \frac{\exp(\epsilon/2) + 3}{6(\exp(\epsilon/2) - 1)^2} \approx 256.7.
\end{aligned} \tag{6.27}$$

Similar to Equation 6.26, we have:

$$\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*} - \boldsymbol{w}^{\star}\right\|^2\right) - \mathbb{E}\left(\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{\star}\right\|^2\right) \\
=& -2\left\langle \boldsymbol{w}^t - \eta^t \nabla^t, \mathbf{0}\right\rangle + C'^2 \eta^{t\,2} \bar{p}_k^2 N^2 \sum_{z=1}^{785} {\boldsymbol{\mu}_z^t}^2 \\
&+ C'^2 \eta^{t\,2} \bar{p}_k^2 N \sum_{z=1}^{785} {\boldsymbol{\sigma}_z^t}^2 + 2\left\langle \boldsymbol{w}^{\star}, \mathbf{0}\right\rangle \approx 10,542,640.
\end{aligned} \tag{6.28}$$

Gaussian mechanism. Based on [30], we set $\delta = 10^{-7}$ and derive the corresponding expectation and variance of the normalized noise:

$$\begin{aligned}
\boldsymbol{\mu}_z^t &= \mathbb{E}\left(\boldsymbol{n}_{\nu kz}^t\right) = 0, \\
{\boldsymbol{\sigma}_z^t}^2 &= \mathbb{E}\left({\boldsymbol{n}_{\nu kz}^t}^2\right) - \mathbb{E}^2\left(\boldsymbol{n}_{\nu kz}^t\right) = 4\ln\frac{1.25}{\delta}/\epsilon^2 \approx 4,694.5.
\end{aligned} \tag{6.29}$$

Accordingly, we have:

$$\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1*}-\boldsymbol{w}^{\star}\right\|^2\right)-\mathbb{E}\left(\left\|\boldsymbol{w}^{t+1}-\boldsymbol{w}^{\star}\right\|^2\right) \\
=&-2\left\langle \boldsymbol{w}^t-\eta^t\nabla^t,\mathbf{0}\right\rangle+C^2{\eta^t}^2\bar{p}_k^{-2}N^2\sum_{z=1}^{785}{\boldsymbol{\mu}_z^t}^2 \\
&+C^2{\eta^t}^2\bar{p}_k^{-2}N\sum_{z=1}^{785}{\boldsymbol{\sigma}_z^t}^2+2\left\langle \boldsymbol{w}^{\star},\mathbf{0}\right\rangle\approx 170,859.
\end{aligned} \tag{6.30}$$

Recall from Section 6.1.3 that a smaller efficiency difference means better utility. Therefore, our benchmark reveals that Gaussian provides the best performance (i.e., $170,859$), followed by Piecewise (i.e., $10,542,640$), and then Gaussian (i.e., $25,521,314$).

This case study confirms a highly-recognized perception that Gaussian performs far better than the other mechanisms in high-dimensional tasks [103]. Still, Laplace and Piecewise, which provide stronger privacy preservation, are much attractive to participants of federated training. Motivated by this, we next present a geometric perturbation strategy to enhance efficiencies of federated LDP-SGD, without changing the LDP mechanism itself.

6.2 LDPVec: Enhancing Federated LDP-SGD from A Geometric Perspective

In our framework, we observe that the direction of noise (see Item B) and the magnitude of noise (see Item C) jointly impact the model efficiency. While Item C can be reduced by fine-tuning the clipping threshold and the learning rate, same techniques are less capable of counteracting the impact of Item B because of the inner product between noise μ_ν^t and the original model $\boldsymbol{w}^t-\eta^t\nabla^t$. As a result, the direction of a gradient is seriously obfuscated, which is the key reason of low model efficiency. While only the direction of a gradient is essential for performing local SGD, we seize the opportunity to **only perturb the direction of a gradient, rather than the**

whole gradient as LDP-SGD, so that the noise on the direction of a gradient can be directly reduced. Instead of only perturbing the local gradient as numeric values in existing works, our strategy, without changing the underlying LDP mechanism, only perturbs and the direction of a local gradient to preserver better directional information while still maintaining LDP scheme. As such, this strategy is orthogonal to all existing LDP mechanisms and LDP-SGD optimizations.

In what follows, we first introduce d-spherical coordinate system [126], where a d-dimensional gradient is converted to one magnitude and one direction. By perturbing the directions of local gradients in such a system, we propose our perturbation strategy *LDPVec* to enhance the global model efficiency. Rigorous analysis is provided to prove its compliance with LDP definition and huge advantages over numerical perturbation.

6.2.1 LDPVec—Vectorized Perturbation for Federated LDP-SGD

While existing works implement numerical perturbation in the rectangular coordinate system, our perturbation strategy *LDPVec* instead mitigates the negative impact of LDP via d-spherical coordinate system. As shown in Algorithm 6, *LDPVec* perturbs local gradients in five steps. First, the clipped local gradient is converted into a d-spherical coordinate system. Afterwards, the noise multiplier $\boldsymbol{n}_\nu$, which determines the level of privacy preservation, is derived as per privacy budget ϵ, probability δ and the mechanism itself $\mathcal{M}$.

To directly control the noise on the direction, we introduce **Angle Clipping** $\beta\,\{0 < \beta \leq 1\}$ in Step 5, which bounds the range of the direction. Accordingly, the range of angles in $1 \leq z \leq d-2$-th dimension is $\beta\pi$ while that in $z = d-1$-th dimension is $2\beta\pi$. The rationale between this is that any angle $\boldsymbol{\theta}_z$ of gradients' directions from SGD is usually concentrated in a certain range, rather than uniformly distributed in the whole direction space, i.e., $\{-\pi, \pi\}$. As such, preserving the direction in the whole space is

actually over-protective and therefore not always optimal. By this angle clipping, we can reduce the sensitivity of respective mechanism and continue noise addition.

In terms of the sensitivity $\Delta\boldsymbol{\theta}$ of the direction $\boldsymbol{\theta}$, it is divided into two cases. As for strict LDP mechanisms (e.g., Laplace and Piecewise), $\Delta\boldsymbol{\theta} = \beta(d-2)(\pi-0) + (\pi-(-\pi)) = \beta d\pi$ because of the use of L_1-sensitivity [103, 134]. As for relaxed LDP mechanisms (e.g., Gaussian), $\Delta\boldsymbol{\theta} = \beta\sqrt{(d-2)(\pi-0)^2 + (\pi-(-\pi))^2} = \beta\sqrt{d+2}\pi$ due to the use of L_2-sensitivity [103]. As a result, the noise scale for preserving each angle is $\Delta\boldsymbol{\theta}\boldsymbol{n}_\nu$. As for the magnitude, $\left\|\tilde{\boldsymbol{g}}_k^t\right\|$ from the clipped gradient $\tilde{\boldsymbol{g}}_k^t$ actually does not contain original information. As such, $\left\|\tilde{\boldsymbol{g}}_k^t\right\|$ is not necessary to preserve. Finally, we convert noise-free magnitude and perturbed direction back to rectangular-coordinate system to derive the perturbed gradient $\boldsymbol{g}_k^\star$ (Step 9). **Notably, we only change the way to perturb the gradient, while the training process remains the same as existing works (see Section 6.1.1).**

The main challenges lie in two folds. First, whether LDPVec follows LDP definition requires further confirmation. Second, the rationale behind LDPVec still needs interpretation. To address the first challenge, we establish the following theorem to confirm that the conversion between coordinate systems does not change the LDP preservation.

Algorithm 6 LDPVec

Input: the LDP mechanism $\mathcal{M}$, local clipped gradient $\tilde{\boldsymbol{g}}_k^t$, privacy budget ϵ and probability δ, bounding factor β.

Output: perturbed gradient $\boldsymbol{g}_k^{t\star}$.

1: Convert $\tilde{\boldsymbol{g}}_k^t$ to d-spherical coordinates as $\left(\left\|\tilde{\boldsymbol{g}}_k^t\right\|, \boldsymbol{\theta}_k\right)$, according to Equation **??** and Equation **??**.

2: Derive the noise multiplier $\boldsymbol{n}_\nu$ as per ϵ, δ and $\mathcal{M}$.

3: Bound the privacy region Δ of $\boldsymbol{\theta}$ as follows:

$$\Delta\boldsymbol{\theta}_z = \begin{cases} \Delta\boldsymbol{\theta}_{1\leq z\leq d-2} & = \beta\pi, \\ \Delta\boldsymbol{\theta}_{d-1} & = 2\beta\pi. \end{cases}$$

4: Add noise to the original direction as follows:

$$\boldsymbol{\theta}_k^* = \boldsymbol{\theta}_k + \boldsymbol{n}_k = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}\boldsymbol{n}_\nu.$$

where

$$\Delta\boldsymbol{\theta} = \begin{cases} \beta d\pi & \text{for strict LDP mechanisms,} \\ \beta\sqrt{d+2}\pi & \text{for relaxed LDP mechanisms .} \end{cases}$$

5: Convert $\left(\left\|\tilde{\boldsymbol{g}}_k^t\right\|, \boldsymbol{\theta}_k^*\right)$ back to rectangular coordinates as the perturbed gradient $\boldsymbol{g}_k^{t\star}$, according to Equation **??**.

Theorem 14. *Given any LDP mechanism $\mathcal{M}$ and $\boldsymbol{g} \leftrightarrow (\|\boldsymbol{g}\|, \boldsymbol{\theta})$, $\boldsymbol{g}^\star$ of LDPVec is (ϵ, δ)-locally differentially private if $\boldsymbol{\theta}^*$ are (ϵ, δ)-locally differentially private.*

Proof. For any three pairs of $\boldsymbol{g}_i, \boldsymbol{g}_j$, $\|\boldsymbol{g}\|_i, \|\boldsymbol{g}\|_j$ and $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j$, we have:

$$\begin{aligned} &\Pr(\mathcal{M}(\boldsymbol{g}_i) = \boldsymbol{g}^\star) \\ =&\Pr\left(\|\boldsymbol{g}\|_i = \|\boldsymbol{g}\|, \mathcal{M}(\boldsymbol{\theta}_i) = \boldsymbol{\theta}^*\right) \\ =&\Pr\left(\|\boldsymbol{g}\|_i = \|\boldsymbol{g}\|\right)\Pr\left(\mathcal{M}(\boldsymbol{\theta}_i) = \boldsymbol{\theta}^*\right) \end{aligned} \tag{6.31}$$

Note that $\Pr\left(\|\boldsymbol{g}\|_i = \|\boldsymbol{g}\|\right) = \Pr(\|\boldsymbol{g}\|_j = \|\boldsymbol{g}\|)$. By applying this to Equation 6.31, we

have:

$$\begin{aligned}\Pr(\mathcal{M}(\boldsymbol{g}_i) = \boldsymbol{g}^\star) \leq & \Pr(\|\boldsymbol{g}\|_j = \|\boldsymbol{g}\|) \Pr\left(\mathcal{M}(\boldsymbol{\theta}_j) = \boldsymbol{\theta}^*\right) \\ & e^\epsilon + \delta \\ \leq & e^\epsilon \Pr(\mathcal{M}(\boldsymbol{g}_j) = \boldsymbol{g}^\star) + \delta\end{aligned} \tag{6.32}$$

by which this lemma is proven. □

As for the superiority of LDPVec to traditional LDP on federated SGD, the introduction of hyper-spherical coordinate system has two advantages. First, LDPVec prevents noise accumulation on the direction of a gradient. For example, we have a three-dimensional gradient $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3)$. Following traditional LDP, these three should be added noise $\boldsymbol{n} = (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)$. For the direction of this perturbed gradient $\boldsymbol{\theta}$, its first angle $\boldsymbol{\theta}_1$ should be $\arctan2\left(\sqrt{(\boldsymbol{g}_2 + \boldsymbol{n}_2)^2 + (\boldsymbol{g}_3 + \boldsymbol{n}_3)^2}, \boldsymbol{g}_1 + \boldsymbol{n}_1\right)$, according to Equation 4. It is very obvious that noise of three dimensions $(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)$ is accumulated to the first angle $\boldsymbol{\theta}_1$. As such, numerical perturbation of LDP seriously degrades the accuracy of directional information. LDPVec otherwise independently controls the noise on each angle and therefore prevents noise accumulation. Second, via coordinates conversion, d-dimensional gradient is transferred to one magnitude and $d-1$ directions. Afterwards, we allocate all privacy budgets to angles, which can better preserves directional information. Third, LDPVec, via hyper-spherical coordinates, can directly control the noise added to the direction, while traditional LDP cannot. In specific, "directly control" is reflected in two aspects, i.e., the noise multiplier $\boldsymbol{n}_\nu$ and sensitivity $\Delta\boldsymbol{\theta}$. Controlling both factors in terms of direction perturbation, LDPVec preserves more accurate directional information of gradients, thus providing better trade-off between efficiency and privacy.

6.3 Experimental Evaluation

In this section, we compare *LDPVec* with conventional numerical perturbation (denoted as *DP*) under three state-of-the-art LDP mechanisms in federated LDP-SGD, namely, Laplace, Piecewise and Gaussian. In addition, we also implement noise-free federated SGD (denoted as *No noise*) as a baseline to demonstrate the feasibility of *LDPVec* in practice. Note that although LDPVec is a brand-new mechanism, most existing optimizations on LDP-SGD, such as [70, 86, 141], can also be applied to LDPVec. Therefore, for fair comparison, we do not implement such optimizations for LDP-SGD.

6.3.1 Datasets and Models

We use two common benchmark datasets for federated LDP-SGD — MNIST and CIFAR-10.

MNIST. This is a dataset of 70,000 gray-scale images (28x28 pixels) of handwritten digits from 0 to 9, commonly used for training and testing machine learning algorithms in image recognition tasks. As one of the de-facto benchmark for evaluating the performance of algorithms, it consists of 60,000 training images and 10,000 testing images, with an even distribution across the 10 digit classes.

CIFAR-10. It is a dataset of 60,000 small (32x32 pixels) color images, divided into 10 distinct classes such as animals and vehicles, used for machine learning and computer vision tasks. It contains 50,000 training images and 10,000 testing images, with each class having an equal number of images.

As with the settings in Section 6.1.4, both datasets are distributed to each local device in a non-iid fashion where each device contains samples of only two classes. We adopt two machine learning models and one deep learning models as local models, namely, the logistic regression (LR) [150], multilayer perceptron (MLP) [12] and Convolu-

tional Neural Network (CNN) [130]. LR is implemented in MATLAB and applied to MNIST, while MLP and CNN are implemented in Python and applied to CIFAR-10. Their architectures and parameters are listed in Table 6.1, Table 6.2 and Table 6.3, respectively. The loss function in both models is cross-entropy loss.

Table 6.1: Parameters for Logistic Regression on MNIST

Layer	Weights	Biases	Total
Output	784	1	785

Table 6.2: MLP Architecture and Parameters on CIFAR-10.

Layer	Description	Number of Parameters
Input	3072 dimensions (32x32x3)	0
Hidden	200 neurons, tanh activation	$(3072 + 1) \times 200 = 614600$
Output	10 classes (CIFAR-10)	$(200 + 1) \times 10 = 2010$
Total		$614600 + 2010 = 616{,}610$

Table 6.3: CNN Architecture and Parameters on CIFAR-10.

Layer	**Parameters**
Convolutional Layer1	$3 \times 5 \times 5 \times 10$ (weights) + 10 (biases)
Convolutional Layer2	$10 \times 5 \times 5 \times 20$ (weights) + 20 (biases)
Fully Connected Layer1	$8 \times 8 \times 20 \times 50$ (weights) + 50 (biases)
Fully Connected Layer2	50×10 (weights) + 10 (biases)
Total Parameters:	$760 + 5020 + 64250 + 510 = 70,540$

6.3.2 Parameter Settings for Federated LDP-SGD

In our experiments, we vary the privacy budget allocated to each round of federated LDP-SGD $\epsilon \in \{1.5, 4.9, 15.3, 48.5\}$, and the number of local devices $N \in \{500, 1000\}$. The total number of iterations in machine learning tasks is varied in the set $\{500, 5000\}$ while the number of local iterations in each training round E is varied in the set $\{20, 50\}$, respectively. In terms of CNN, we set the total epochs and local epochs in each training round 100 and 10, respectively. To evaluate our strategy objectively, we use the basic SGD without any special optimization technique (e.g., momentum). The batch size B is fixed to 32, and the leaning rate is decayed by the following scheme $\eta^t = \frac{\eta^0}{1+t}$, where η^0 is chosen from the set $\{0.1, 0.01\}$. Following [164], we fix $C = 0.1$ and $\delta = 10^{-5}$.

As for privacy budget partitioning, instead of partitioning it as per the number of dimensions, we follow DP-SGD by partitioning it as per the collective sensitivity $\Delta\boldsymbol{\theta}$ [1], which provides better utility than the former.

6.3.3 Effectiveness of LDPVec in Machine Learning

In terms of LR and MLP, we verify the effectiveness of our perturbation strategy *LDPVec*. Due to the simplicity of these models, we do not conduct angle clipping, i.e., $\beta = 1$. In Figure 6.2, we plot the global loss of LR on MNIST dataset, under no perturbation (i.e., blue line), numerical perturbation (i.e., red line), and vectorized perturbation by *LDPVec* (i.e., yellow line). From Figures 6.2 (a)-(i), where the privacy budget in each training round varies from 1.5 to 15.3, we can observe that LDPVec can help federated LDP-SGD to achieve similar performance to noise-free federated SGD. we can observe that our perturbation strategy achieves almost the same performance as noise-free federated SGD in most cases. The only exception is Figure 6.2 (c), where the convergence is delayed by the Laplace noise as the Laplace noise at $\epsilon = 0.00001$ is very large. Nevertheless, *LDPVec* can still converge. On the other hand, the

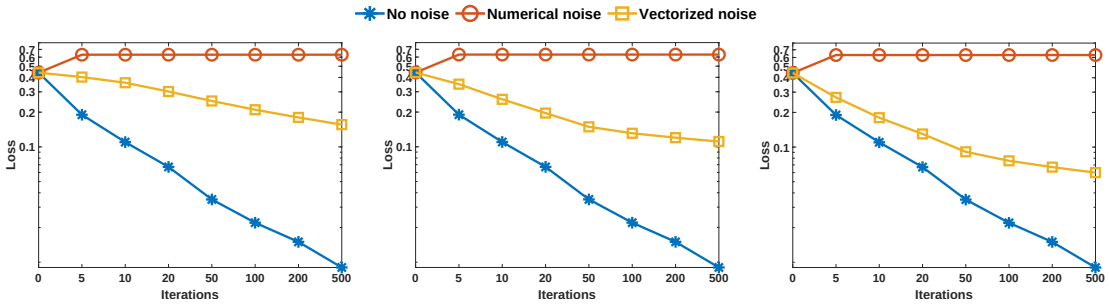
performance of three LDP mechanisms under numerical perturbation is disappointing. In most cases, the training process ceases to converge and deviate from the global optima, especially for strict mechanisms such as Laplace and Piecewise. Even worse, in Figures 6.2 (e), (f), gradient exposure occurs and the whole training process fails. Obviously, these bad situations are also incurred by overwhelming LDP noise from the numerical perturbation. We list the test accuracy of this model without LDP noise in the most left column. As we can see, our perturbation strategy is very feasible and robust even under cases where the numerical perturbation fails.

To test the limit of our strategy, in Figure 6.2 (j)-(k) we set the number of local parities $N = 5$, and $E = 2$ (i.e., noisy aggregation happens every two local iterations). In this extreme case of steeply declined number of local devices and surging occurrences of noise injection, we observe that federated LDP-SGD under the numerical perturbation cannot converge. On the other hand, although *LDPVec* suffers from delayed convergence as in Figure 6.2 (c), the final converged model still achieves almost similar loss to federated SGD without noise.

To demonstrate the generality of *LDPVec* in various machine learning applications, we also conduct similar classification tasks on CIFAR-10 with Multilayer Perceptron (MLP). Due to the extremely large number of parameters, we raise the privacy budget to $\epsilon = 48.5$ per training round and increase the total local iterations to $T = 5,000$. As shown in Figure 6.3, we obtain similar results to LR on MNIST dataset. As such, we can conclude that *LDPVec* can enhance the model efficiency to the degree whereas conventional numerical perturbation may fail.

6.3.4 Effectiveness of LDPVec in Deep Learning

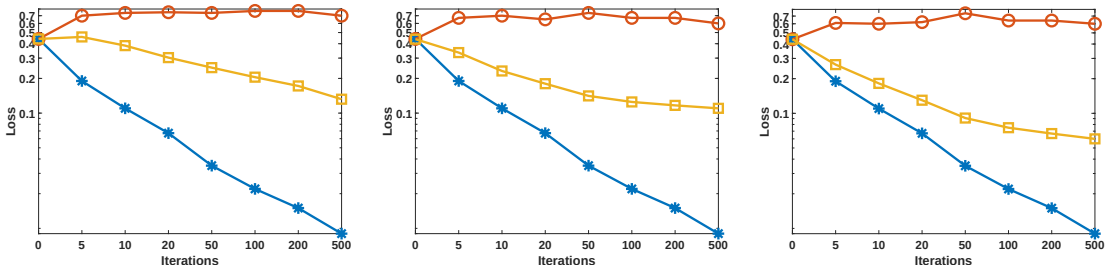
To demonstrate the generality of LDPVec in deep learning, we also conduct experiments on CIFAR-10 with Convolutional Neural Network (CNN). The results are demonstrated in Table 6.4, where suffix "-Vec" denotes our perturbation strategy



(a) Laplace ($\epsilon = 1.5, E = 20, N = 500$)

(b) Laplace ($\epsilon = 4.9, E = 20, N = 500$)

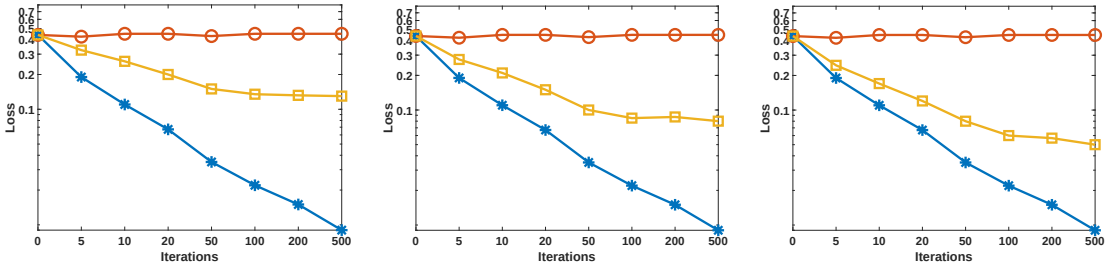
(c) Laplace ($\epsilon = 15.3, E = 20, N = 500$)



(d) Piecewise ($\epsilon = 1.5, E = 20, N = 500$)

(e) Piecewise ($\epsilon = 4.9, E = 20, N = 500$)

(f) Piecewise ($\epsilon = 15.3, E = 20, N = 500$)

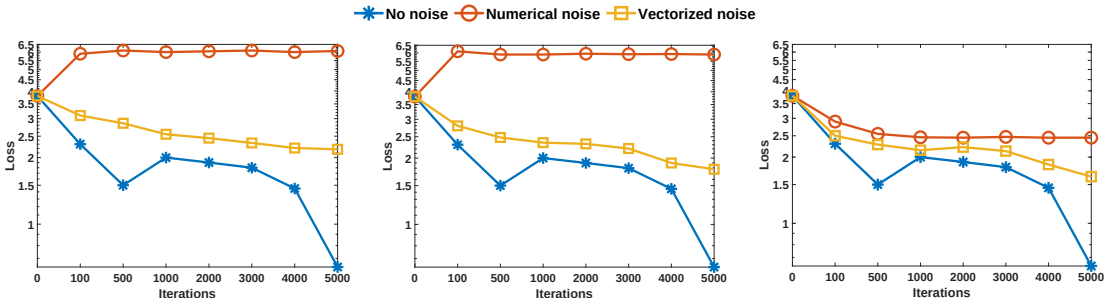


(g) Gaussian ($\epsilon = 1.5, E = 20, N = 500$)

(h) Gaussian ($\epsilon = 4.9, E = 20, N = 500$)

(i) Gaussian ($\epsilon = 15.3, E = 20, N = 500$)

Figure 6.2: Global Loss of Logistic Regression on MNIST under Various ϵ, E and N



(a) Laplace ($\epsilon = 48.5, E = 50, N = 1000$) (b) Piece ($\epsilon = 48.5, E = 50, N = 1000$) (c) Gaussian ($\epsilon = 48.5, E = 50, N = 1000$)

Figure 6.3: Global Loss of Multilayer Perceptron on CIFAR-10

Table 6.4: Perturbation Comparison under Federated CNN: Testing Accuracy ($N = 1000$, $E = 200$, $\beta = 0.05$)

Dataset	Method	$\epsilon = 15.3$	$\epsilon = 48.5$
MNIST (noise-free 95%)	Lap	43.3%	48.2%
	Pie	44.9%	51.4%
	Gau	71.3%	77.3%
	Lap-Vec	72.1%	75.6%
	Pie-Vec	76.3%	79.6%
	Gau-Vec	88.4%	91.7%

and those without it means numerical perturbation. In general, LDPVec outperforms LDP under different privacy budgets when β is 0.05. The reason behind is that LDPVec reduces the original sensitivity (which is extremely large due to a high dimensionality of 70540) to 5%. By bounding the privacy region, the payoff is huge. Not only can Gaussian mechanism achieve better model efficiency, but also Laplace and Piecewise mechanisms can provide usable efficiency while still maintaining strict privacy preservation in practical deep learning tasks.

6.4 Summary

This chapter investigates the efficiencies of various LDP mechanisms in Federated Learning. For model efficiency, we propose a general analytical framework to measure the impact of LDP noise on federated LDP-SGD, which also serve as a benchmark to compare federated LDP-SGD under various LDP mechanisms. For optimization, we propose *LDPVec* to generally enhance the efficiency of federated LDP-SGD while maintaining the same level of LDP protection, without changing LDP mechanism itself. Through theoretical analysis and extensive experiments, we confirm the generality and effectiveness of our framework and perturbation strategy under two commonly-used benchmark datasets and three prevalent models.

For the future work, we plan to extend *LDPVec* to more complicated networks, such as ResNet.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

This thesis comprehensively discusses the application of differential privacy (DP) in data mining techniques, specifically focusing on mean estimation in high-dimensional spaces, stochastic gradient descent (SGD), and federated learning. Below is a detailed conclusion for each research area:

7.1.1 Mean Estimation in High-Dimensional Spaces by LDP Mechanisms

Our research investigates the utilities of mean estimation by LDP mechanisms in high-dimensional spaces and presents a general toolbox *LDPTube*. *LDPTube* proposed an analytical framework that evaluates any LDP mechanism based on the deviation between the estimated and the true mean. Additionally, *LDPTube* introduced *HDR4ME**, a re-calibration protocol to enhance the utility of aggregation results from these LDP mechanisms. Through theoretical analysis and extensive experiments, we confirmed the generality and effectiveness of our framework and re-calibration proto-

col.

7.1.2 Optimization of DP-SGD

We optimize DP-SGD from a novel perspective by analyzing the impact of DP noise on the training process of SGD. Our findings revealed that the perturbations introduced are sub-optimal as they bias the direction of updates. This insight led us to propose the *GeoDP* mechanism, which reduces noise directionality to improve model efficiency. Both rigorous proofs and experimental validations confirmed the effectiveness and generality of GeoDP.

7.1.3 Efficiencies of LDP Mechanisms in Federated Learning

Our investigation focused on proposing a general analytical framework to measure the impact of LDP noise on federated LDP-SGD. We also introduced *LDPVec*, a strategy to enhance the efficiency of federated LDP-SGD while maintaining the same level of LDP protection. Our analyses confirm the generality and effectiveness of our approach.

7.2 Future Work

The future work for these research areas includes several key expansions:

- **Extending Analytical Frameworks:** We aim to broaden the applicability of our frameworks to other data types and incorporate more data analysis tasks.
- **Exploring Dimensional Extents:** Plans include expanding research to both multi-dimensional and one-dimensional spaces.

- **Integrating Training Optimizations:** We plan to study the effects of mainstream training optimizations (e.g., regularization) on the performance of our proposed mechanisms.
- **Applying to Complex Network Architectures:** We aim to extend strategies like *LDPVec* to more complex network architectures, such as ResNet.

This future work will contribute significantly to enhancing the privacy-security of data mining techniques, marking substantial advancements in the robustness and adaptability of differential privacy applications.

References

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

[2] Mário Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. Local differential privacy on metric spaces: optimizing the trade-off with utility. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 262–267. IEEE, 2018.

[3] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

[4] Sean Augenstein, H Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

[5] Ergute Bao, Yin Yang, Xiaokui Xiao, and Bolin Ding. Cgm: an enhanced mechanism for streaming data collection with local differential privacy. *Proceedings of the VLDB Endowment*, 14(11):2258–2270, 2021.

[6] Ergute Bao, Yizheng Zhu, Xiaokui Xiao, Yin Yang, Beng Chin Ooi, Benjamin Hong Meng Tan, and Khin Mi Mi Aung. Skellam mixture mechanism: a novel approach to federated learning with differential privacy. *Proceedings of the VLDB Endowment*, 15(11):2348–2360, 2022.

[7] Manuel Barbosa, Sonia Ben Mokhtar, Pascal Felber, Francisco Maia, Miguel Matos, Rui Oliveira, Etienne Riviere, Valerio Schiavoni, and Spyros Voulgaris. Safethings: Data security by design in the iot. In *EDCC*, pages 117–120, 2017.

[8] Raef Bassily. Linear queries estimation with local differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 721–729. PMLR, 2019.

[9] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *arXiv preprint arXiv:1707.04982*, 2017.

[10] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing (STOC)*, pages 127–135, 2015.

[11] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*. Springer, 2012.

[12] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[13] Franziska Boenisch, Christopher Mühl, Adam Dziedzic, Roy Rinberg, and Nicolas Papernot. Have it your way: Individualized privacy assignment for dp-sgd. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Dmytro Bogatov, Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. εpsolute: Efficiently querying databases while providing differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2262–2276, 2021.

[15] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[16] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[17] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. *Advances in Neural Information Processing Systems*, 36, 2024.

[18] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[19] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

[20] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security)*, pages 2633–2650, 2021.

[21] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 82–102. Springer, 2013.

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

[23] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, 4(11):1087–1098, 2011.

[24] Xiangyi Chen, Zhiwei Steven Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: a geometric perspective. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020.

[25] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pages 1655–1658, 2018.

[26] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Marginal release under local differential privacy. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pages 131–146. ACM, 2018.

[27] Jorge Cortés, Geir E. Dullerud, Shuo Han, Jerome Le Ny, Sayan Mitra, and George J. Pappas. Differential privacy in control and network systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4252–4272, 2016.

[28] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *arXiv preprint arXiv:1712.01524*, 2017.

[29] Rong Du, Qingqing Ye, Yue Fu, and Haibo Hu. Collecting high-dimensional and correlation-constrained data with local differential privacy. In *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2021.

[30] Jiawei Duan, Qingqing Ye, and Haibo Hu. Utility analysis and enhancement of ldp mechanisms in high-dimensional space. In *ICDE*, pages 407–419. IEEE, 2022.

[31] Jiawei Duan, Qingqing Ye, Haibo Hu, and Xinyue Sun. Ldptube: Theoretical utility benchmark and enhancement for ldp mechanisms in high-dimensional space. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[32] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 429–438. IEEE, 2013.

[33] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[34] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[35] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

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

[37] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2), 2010.

[38] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014*

ACM SIGSAC conference on computer and communications security (CCS), pages 1054–1067, 2014.

[39] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)*, pages 211–222, 2003.

[40] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3):41–61, 2016.

[41] Victor AE Farias, Felipe T Brito, Cheryl Flynn, Javam C Machado, Subhabrata Majumdar, and Divesh Srivastava. Local dampening: Differential privacy for non-numeric queries via local sensitivity. *the VLDB Journal*, 32(6):1191–1214, 2023.

[42] Yuqing Feng, Peter Kairouz, Lalitha Sankar, and Ram Rajagopal. Privacy amplification by iteration. In *Advances in Neural Information Processing Systems*, 2020.

[43] Hans Fischer. *A history of the central limit theorem. From classical to modern probability theory*. 01 2011.

[44] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[45] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX security symposium (USENIX Security 14)*, pages 17–32, 2014.

[46] Jie Fu, Qingqing Ye, Haibo Hu, Zhili Chen, Lulu Wang, Kuncan Wang, and Ran Xun. Dpsur: Accelerating differentially private stochastic gradient descent using selective update and release. *arXiv preprint arXiv:2311.14056*, 2023.

[47] Dawei Gao, Daoyuan Chen, Zitao Li, Yuexiang Xie, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. Fs-real: A real-world cross-device federated learning platform. *Proceedings of the VLDB Endowment*, 16(12):4046–4049, 2023.

[48] Jason Ge, Zhaoran Wang, Mengdi Wang, and Han Liu. Minimax-optimal privacy-preserving sparse pca in distributed systems. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018*, 2018.

[49] Quan Geng, Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1176–1184, 2015.

[50] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[51] Sameera Ghayyur, Yan Chen, Roberto Yus, Ashwin Machanavajjhala, Michael Hay, Gerome Miklau, and Sharad Mehrotra. Iot-detective: Analyzing iot data under differential privacy. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pages 1725–1728, 2018.

[52] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[53] Neil Zhenqiang Gong and Bin Liu. Attribute inference attacks in online social networks. *ACM Transactions on Privacy and Security (TOPS)*, 21(1):1–30, 2018.

[54] Xueluan Gong, Yanjiao Chen, Wenbin Yang, Guanghao Mei, and Qian Wang. Inversenet: Augmenting model extraction attacks with training data inversion. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2439–2447, 2021.

[55] Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems (NIPS)*, 34:11631–11642, 2021.

[56] Xiaolan Gu, Ming Li, Yueqiang Cheng, Li Xiong, and Yang Cao. PCKV: locally differentially private correlated key-value data collection with optimized utility. In *USENIX Security Symposium*, 2020.

[57] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 2019.

[58] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[59] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.

[60] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[61] Mikko Heikkilä, Eemil Lagerspetz, Samuel Kaski, Kana Shimizu, Sasu Tarkoma, and Antti Honkela. Differentially private bayesian learning on distributed data. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[62] Stella Ho, Youyang Qu, Bruce Gu, Longxiang Gao, Jianxin Li, and Yong Xiang. Dp-gan: Differentially private consecutive data publishing using generative adversarial nets. *Journal of Network and Computer Applications*, 185:103066, 2021.

[63] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(9), 2004.

[64] Xueyang Hu, Mingxuan Yuan, Jianguo Yao, Yu Deng, Lei Chen, Qiang Yang, Haibing Guan, and Jia Zeng. Differential privacy in telco big data platform. *Proceedings of the VLDB Endowment*, 8(12):1692–1703, 2015.

[65] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[66] Jinyuan Jia and Neil Zhenqiang Gong. Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In *27th USENIX Security Symposium (USENIX Security)*, pages 513–529, 2018.

[67] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR, 2016.

[68] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[69] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *The Journal of Machine Learning Research*, 17(1):492–542, 2016.

[70] Muah Kim, Onur Günlü, and Rafael F Schaefer. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2650–2654. IEEE, 2021.

[71] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[72] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[73] V. Yu. Korolev and I. G. Shevtsova. On the upper bound for the sbsolute constant in the eerry–esseen inequality. *Theory of Probability & Its Applications*, 54(4):638–658, 2010.

[74] Samuel Kotz, Tomasz Kozubowski, and Krzysztof Podgórski. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Number 183. Springer Science & Business Media, 2001.

[75] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[76] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[77] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

[78] Don S Lemons. An introduction to stochastic processes in physics, 2003.

[79] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems (NIPS)*, 28:379–387, 2015.

[80] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[81] Xianxian Li, Chunfeng Luo, Peng Liu, and Li-e Wang. Information entropy differential privacy: A differential privacy protection data method based on rough set theory. In *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress*, pages 918–923. IEEE, 2019.

[82] Zitao Li, Bolin Ding, Ce Zhang, Ninghui Li, and Jingren Zhou. Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment*, 15(4), 2021.

[83] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Škoric. Estimating numerical distributions under local differential privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 621–635, 2020.

[84] Jianqing Liu, Chi Zhang, and Yuguang Fang. Epic: A differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet of Things Journal*, 5(2):1206–1217, 2018.

[85] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. Projected federated averaging with heterogeneous differential privacy. *Proceedings of the VLDB Endowment*, 15(4):828–840, 2021.

[86] Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated sgd under local differential privacy with top-k dimension selection. In *International Conference on Database Systems for Advanced Applications (DASFAA)*, page 485–501, Berlin, Heidelberg, 2020. Springer-Verlag.

[87] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. {ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4525–4542, 2022.

[88] Zhaobin Liu, Zhiyi Huang, Haoze Lyu, Zhiyang Li, Weijiang Liu, et al. Dynapro: Dynamic wireless sensor network data protection algorithm in iot via differential privacy. *IEEE Access*, 7:167754–167765, 2019.

[89] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[90] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[91] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706, 2019.

[92] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017*, pages 263–275. IEEE, 2017.

[93] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 1–15, 2018.

[94] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[95] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[96] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.

[97] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, Bin Yu, et al. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical science*, 27(4):538–557, 2012.

[98] Thông T Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.

[99] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. *Advances in Neural Information Processing Systems (NIPS)*, 27:1574–1582, 2014.

[100] Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 121–144, 2019.

[101] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

[102] Christian Peukert, Stefan Bechtold, Michail Batikas, and Tobias Kretschmer. Regulatory spillovers and data governance: Evidence from the gdpr. *Marketing Science*, 2022.

[103] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

[104] Ismini Psychoula, Liming Chen, and Oliver Amft. Privacy risk awareness in wearables and the internet of things. *IEEE Pervasive Computing*, 19(3):60–66, 2020.

[105] 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 (FOCS)*, pages 531–540, 2008.

[106] Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A McCann, and S Yu Philip. Lopub: High-dimensional crowdsourced data publication with local differential privacy. *IEEE Transactions on Information Forensics and Security (TIFS)*, 13(9):2151–2166, 2018.

[107] Herbert Robbins and Sutton Monro. Stochastic gradient descent. *Journal of the American Statistical Association*, 1951.

[108] Wenqiang Ruan, Mingxin Xu, Wenjing Fnag, Li Wang, Lei Wang, and Weili Han. Private, efficient, and accurate: Protecting models trained by multi-party learning with differential privacy. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 76–93. IEEE Computer Society, 2022.

[109] David E Rumelhart, James L McClelland, and PDP Research Group. *Parallel distributed processing: explorations in the microstructure of cognition*. MIT Press, 1986.

[110] 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*, 2019.

[111] Tom Sander, Pierre Stock, and Alexandre Sablayrolles. Tan without a burn: Scaling laws of dp-sgd. In *International Conference on Machine Learning*, pages 29937–29949. PMLR, 2023.

[112] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[113] Christine Schäler, Thomas Hütter, and Martin Schäler. Benchmarking the utility of w-event differential privacy mechanisms-when baselines become mighty competitors. *Proceedings of the VLDB Endowment*, 16(8):1830–1842, 2023.

[114] Mohamed Seif, Ravi Tandon, and Ming Li. Wireless federated learning with local differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2604–2609. IEEE, 2020.

[115] J. George Shanthikumar and Ushio Sumita. A central limit theorem for random sums of random variables. *Operations Research Letters*, 3(3):153–155, 1984.

[116] Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

[117] Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer, 2018.

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

[119] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[120] Jordi Soria-Comas and Josep Domingo-Ferrer. Optimal data-independent noise for differential privacy. *Information Sciences*, 250:200–214, 2013.

[121] Haipei Sun, Xiaokui Xiao, Issa Khalil, Yin Yang, Zhan Qin, Hui Wendy Wang, and Ting Yu. Analyzing subgraph statistics from extended local views with decentralized differential privacy. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 703–717. ACM, 2019.

[122] Xinyue Sun, Qingqing Ye, Haibo Hu, Jiawei Duan, Qiao Xue, Tianyu Wo, and Jie Xu. Puts: Privacy-preserving and utility-enhancing framework for trajectory synthesization. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[123] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[124] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.

[125] Qiaoyue Tang, Frederick Shpilevskiy, and Mathias Lécuyer. Dp-adambc: Your dp-adam is actually dp-sgd (unless you apply bias correction). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15276–15283, 2024.

[126] George B. Jr. Thomas and Maurice D. Weir. *Multivariable Calculus and Linear Algebra*. Pearson/Addison-Wesley, 2006.

[127] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[128] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.

[129] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data*, pages 2587–2596. IEEE, 2019.

[130] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692, 2015.

[131] Chandan S Vora. On the estension of lipschitz functions with respect to two hilbert norms and two lipschitz conditions. *Rendiconti del Seminario Matematico della Università di Padova*, 50:173–183, 1973.

[132] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 36–52. IEEE, 2018.

[133] Di Wang and Jinhui Xu. Principal component analysis in the local differential privacy model. *Theoretical Computer Science*, 809:296–312, 2020.

[134] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 638–649. IEEE, 2019.

[135] Shaowei Wang, Liusheng Huang, Yiwen Nie, Pengzhan Wang, Hongli Xu, and Wei Yang. Privset: Set-valued data analyses with locale differential privacy. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1088–1096. IEEE, 2018.

[136] Shaowei Wang, Yuqiu Qian, Jiachun Du, Wei Yang, Liusheng Huang, and Hongli Xu. Set-valued data publication with local privacy: tight error bounds and efficient mechanisms. *Proceedings of the VLDB Endowment*, 13(8):1234–1247, 2020.

[137] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *USENIX Security Symposium*, pages 729–745, 2017.

[138] Tianhao Wang, Ninghui Li, and Somesh Jha. Locally differentially private frequent itemset mining. In *the Symposium on Security and Privacy (S&P)*, pages 127–143. IEEE, 2018.

[139] Tingting Wang, Shixun Huang, Zhifeng Bao, J Shane Culpepper, Volkan Dedeoglu, and Reza Arablouei. Optimizing data acquisition to enhance machine learning performance. *Proceedings of the VLDB Endowment*, 17(6):1310–1323, 2024.

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

[141] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security (TIFS)*, 15:3454–3469, 2020.

[142] Tianyu Xia, Shuheng Shen, Su Yao, Xinyi Fu, Ke Xu, Xiaolong Xu, and Xing Fu. Differentially private learning with per-sample adaptive clipping. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pages 10444–10452, 2023.

[143] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[144] Yonghui Xiao, Li Xiong, Si Zhang, and Yang Cao. Loclok: Location cloaking with differential privacy via hidden markov model. *Proceedings of the VLDB Endowment*, 10(12):1901–1904, 2017.

[145] Naili Xing, Shaofeng Cai, Gang Chen, Zhaojing Luo, Beng Chin Ooi, and Jian Pei. Database native model selection: Harnessing deep neural networks in database systems. *Proceedings of the VLDB Endowment*, 17(5):1020–1033, 2024.

[146] Lijie Xu, Shuang Qiu, Binhang Yuan, Jiawei Jiang, Cedric Renggli, Shaoduo Gan, Kaan Kara, Guoliang Li, Ji Liu, Wentao Wu, et al. Stochastic gradient descent without full data shuffle: with applications to in-database machine learning and deep learning systems. *The VLDB Journal*, pages 1–25, 2024.

[147] Min Xu, Bolin Ding, Tianhao Wang, and Jingren Zhou. Collecting and analyzing data jointly from multiple services under local differential privacy. *Proceedings of the VLDB Endowment*, 13(12):2760–2772, 2020.

[148] Min Xu, Tianhao Wang, Bolin Ding, Jingren Zhou, Cheng Hong, and Zhicong Huang. Dpsaas: Multi-dimensional data sharing and analytics as services under local differential privacy. *Proceedings of the VLDB Endowment*, 12(12):1862–1865, 2019.

[149] Qiao Xue, Youwen Zhu, and Jian Wang. Mean estimation over numeric data with personalized local differential privacy. *Frontiers of Computer Science*, 16(3):1–10, 2022.

[150] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[151] Qingqing Ye and Haibo Hu. Local differential privacy: Tools, challenges, and opportunities. In *WISE*, pages 13–23. Springer, 2020.

[152] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. Lf-gdpr: A framework for estimating graph metrics with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[153] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. Towards locally differentially private generic graph metric estimation. In *2020 IEEE 36th International Conference on Data Engineering(ICDE)*, pages 1922–1925. IEEE, 2020.

[154] Qingqing Ye, Haibo Hu, Ninghui Li, Xiaofeng Meng, Huadi Zheng, and Haotian Yan. Beyond value perturbation: local differential privacy in the temporal setting. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10. IEEE, 2021.

[155] Qingqing Ye, Haibo Hu, Xiaofeng Meng, and Huadi Zheng. PrivKV: Key-value data collection with local differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 317–331. IEEE, 2019.

[156] Qingqing Ye, Haibo Hu, Xiaofeng Meng, Huadi Zheng, Kai Huang, Chengfang Fang, and Jie Shi. PrivKVM*: Revisiting key-value statistics estimation with local differential privacy. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2021.

[157] 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)*, pages 268–282, 2018.

[158] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[159] Tiantian Yu, Donglin Bai, and Rui Zhang. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[160] Sepanta Zeighami, Ritesh Ahuja, Gabriel Ghinita, and Cyrus Shahabi. A neural database for differentially private spatial range queries. *Proceedings of the VLDB Endowment*, 15(5):1066–1078, 2022.

[161] Huayi Zhang, Binwei Yan, Lei Cao, Samuel Madden, and Elke Rundensteiner. Metastore: Analyzing deep learning meta-data at scale. *Proceedings of the VLDB Endowment*, 17(6):1446–1459, 2024.

[162] Michael R Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Neural Information Processing Systems (NIPS)*, 2019.

[163] Qiuchen Zhang, Hong kyu Lee, Jing Ma, Jian Lou, Carl Yang, and Li Xiong. Dpar: Decoupled graph neural networks with node-level differential privacy. In *Proceedings of the ACM on Web Conference 2024*, pages 1170–1181, 2024.

[164] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning (ICML)*, 2022.

[165] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

[166] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 212–229. ACM, 2018.

[167] Bo Zhao, Benjamin IP Rubinstein, and Jim Gemmell. Dp-clipper: An improved dp-sgd with gradient clipping adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[168] Huadi Zheng, Qingqing Ye, Haibo Hu, Chengfang Fang, and Jie Shi. BDPL: A boundary differentially private layer against machine learning model extraction attacks. In *ESORICS*, pages 66–83. Springer, 2019.

[169] Huadi Zheng, Qingqing Ye, Haibo Hu, Chengfang Fang, and Jie Shi. Protecting decision boundary of machine learning model with differentially private pertur-

bation. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2020.