University of California
Santa Barbara

Making Differential Privacy Practical via Modern Privacy Accountings and Data-adaptive Analysis

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Yuqing Zhu

Committee in charge:

Professor Yu-Xiang Wang, Chair
Professor William Wang
Professor Lei Li
Dr. Peter Kairouz

December 2023

The Dissertation of Yuqing Zhu is approved.

Professor William Wang

Professor Lei Li

Dr. Peter Kairouz

Professor Yu-Xiang Wang, Committee Chair

October 2023

Making Differential Privacy Practical via Modern Privacy Accountings and Data-adaptive Analysis

by

Yuqing Zhu

To my family and friends.

Acknowledgements

I would first express my sincere gratitude to my advisor, Professor Yu-Xiang Wang. I am very fortunate to be one of your first students. Your are the best PhD advisor I could have hoped for. Your enthusiasm for finding real, meaningful problems in machine learning has changed the way I see research. You are always there for me, explaining complex ideas, always encouraging me to share my own thoughts on our projects, working alongside us late into the night before conference deadlines, and opening my eyes to the amazing world of privacy in machine learning. These past five years have been filled with learning, growing, and moments that I will always cherish.

I would also like to thank my committee member, Dr. Peter Kairouz, for your tremendous support and mentorship throught our collaboration, my internship and my PhD journey. My gratitude also extends to my committee members, Professor William Wang and Professor Lei Li, for their insightful suggestions regarding this thesis and my career path.

Throughout the years, I have had the privilege of working with many brilliant researchers. This thesis would not have been possible without the collaborative efforts of my esteemed collaborators: Xuandong Zhao, Jinshuo Dong, Chong Liu, Rachel Redberg, Jiachen T. Wang, Chendi Wang, Chuan Guo, Weijie Su, Xiang Yu, Manmohan Chandraker, Ruoxi Jia, Prateek Mittal, Yi-Hsuan Tsai, Francesco Pittaluga, Masoud Faraki and Kamalika Chaudhuri.

My heartfelt thanks go to my mentors during my four internships: Xiang Yu, Shanshan Wu, Matthew Joseph, and Huanyu Zhang, for their insightful guidance and support.

I am also grateful for my graduate life in UCSB. My colleagues and friends in UCSB have made this journey even more memorable. A special thanks to Jianyu Xu, Dan Qiao, Xuandong Zhao, Chong Liu, Mengye Liu, Ming Yin, Dheeraj Baby, Rachel Redberg, Kaiqi Zhang, Shiyang Li and Hong Wang and many others, for the wondeful times we shared. I also extend my thanks to my roommates Aiwen Xu, Jinglei Yang and Ling Cai, for the unforgettable moments we have spent together.

Lastly, I owe a debt of gratitude to my family. To my parents, for their unwavering support throughout my life; to my fiancé, Yi Wang, for our eight-year journey and enduring love; and to my cat, Mimi, for bringing endless joy into my life.

Curriculum Vitæ

Yuqing Zhu

Education

2023	Ph.D. in Computer Science, University of California, Santa Barbara.
2018	B.S. in Computer Science, Nanjing University.

Publications

NeurIPS 2023	Jiachen T. Wang, **Yuqing Zhu**, Yu-Xiang Wang, Ruoxi Jia, Prateek Mittal *Threshold KNN-Shapley: A Linear-Time and Privacy-Friendly Approach to Data Valuation.* **Spotlight**.
UAI 2023	**Yuqing Zhu**, Xuandong Zhao, Chuan Guo, and Yu-Xiang Wang. *Private Prediction Strikes Back!" Private Kernelized Nearest Neighbors with Individual Rényi Filter.* **Spotlight**.
AISTATS 2023	Rachel Redberg, **Yuqing Zhu**, Yu-Xiang Wang. *Generalized PTR: User-Friendly Recipes for Data-Adaptive Algorithms with Differential Privacy.* **Oral**.
AISTATS 2022	**Yuqing Zhu**, Yu-Xiang Wang. *Adaptive Private-K-Selection with Adaptive K and Application to Multi-label PATE.*
AISTATS 2022	**Yuqing Zhu**, Jinshuo Dong, Yu-Xiang Wang. *Optimal Accounting of Differential Privacy via Characteristic Function.*
JMLR 2021	Chong Liu, **Yuqing Zhu**, Kamalika Chaudhuri and Yu-Xiang Wang. *Revisiting Model-Agnostic Private Learning: Faster Rates and Active Learning.*
CVPR 2020	**Yuqing Zhu**, Xiang Yu, Manmohan Chandraker, Yu-Xiang Wang. *Private-kNN: Practical Differential Privacy for Computer Vision.*
ICML 2019	**Yuqing Zhu** and Yu-Xiang Wang. *Poisson Subsampled Renyi Differential Privacy.*
TPDP 2020	**Yuqing Zhu**, Chong Liu and Yu-Xiang Wang. *Model-Agnostic Private Learning with Domain Adaptation.* In 2020 CSS Theory and Practice of Differential Privacy Workshop.
FL-NeurIPS-2022	**Yuqing Zhu**, Xiang Yu, Yi-Hsuan Tsai, Francesco Pittaluga, Masoud Faraki, Manmohan chandraker, Yu-Xiang Wang. *Voting-based Approaches For Differentially Private Federated Learning.*

Academic Services

Reviewer	ICML-23, NeurIPS-23, NeurIPS-22, ICML-22, NeurIPS-21, AISTATS-21, ICLR-20, ICML-20, ICML-19, UAI-19, NeurIPS-19

Workshop organizer	PRIML-NeurIPS-21
Program Committee	TPDP-23

Abstract

Making Differential Privacy Practical via Modern Privacy Accountings and
Data-adaptive Analysis

by

Yuqing Zhu

With increasing ethical and legal concerns on privacy in the era of big data, differential privacy (DP) has emerged as the de facto gold standard to disguise membership of individuals with quantiiable privacy guarantee. In DP, the theoretical privacy guarantee directly corresponds to the amount of noise and randomness that must be introduced into a DP mechanism. Therefore, to employ DP algorithms in the real world, it is crucial to develop a tight characterization of privacy analysis. This thesis aims to bridge the gap between theory and DP deployment by refining the constants in privacy guarantees. In the first part of the thesis, we focus on modern privacy accountings, which characterize privacy degradation through fine-grained mechanism-specific analysis, driving much of the recent success in DP deployments. We enhance these modern privacy accountings by generalizing the PLD formalism to handle adaptive composition and amplification by sampling, two foundamental components in the design of DP algorithms. Additionally, we derive nearly optimal bounds for characterizing privacy amplification by sampling in the Rényi DP framework, which directly translate into practical enhancements in private deep learning. In the second part of the thesis, we address the mathematical slack in privacy analysis by incorporating data-adaptive analysis, enabling less noise injection when the input dataset is deemed “nice”.

Contents

Chapter 1

Introduction and background

1.1 Introduction

Machine learning models memorize training data, and this poses privacy risks when training data is sensitive. For example, recent research shows that given query access to GPT-2 models, they are able to recover hundreds of training data points including email address and phone number. As more and more sensitive data is being collected and used, privacy is becoming an increasingly vexing for researchers and policymakers.

The potential privacy issues are not limited to the data release itself but also inferences that can be made about individuals. To formalize privacy requirements, GDPR, the most well-known EU privacy guidance, imposes several mandates for firms to retain personal data. Specifically, GDPR requests "processing of personal data in such a manner that the personal data can no longer be attributed to a specific data", which is also known as data anonymity. To preserve individual privacy, classic approaches involve anonymizing user data by removing personal identifications from the dataset. However, this approach is vulnerable under linkage attacks — where an adversary leverages additional public information to re-identify an anonymized dataset. One example of data anonymization failing is the Netflix competition. Netflix released a user dataset in 2009, where each user data was fully anonymized. Unfortunately, researchers successfully re-identified the Netflix dataset through a cross reference to a publicly available IMDb dataset, see [Narayanan and Shmatikov, 2006].

These concerns force us to confront a fundamental question: How can we gain insights from increasingly massive datasets while providing rigorous privacy guarantees to the individuals whose data we are using?

Differential privacy (DP) [Dwork, 2006] is one of the most promising approaches aims at answering this questions with many good properties; it is a quantifiable and composable definition of privacy that provides provable guarantees against identifications of individuals in a private dataset. Informally, a differentially private mechanism imposes constraints on the extent to which a single data point can influence the output distribution. This is accomplished by introducing randomness into the DP mechanisms (e.g., adding noise), which enables a certain plausible deniability for any individual data in the dataset. The privacy guarantee is parametrized by a parameter ϵ, which measures the maximum impact of one individual's data on the output distribution. In other words, a smaller ϵ leads to more similar output distributions (when any individual data point is added / removed), resulting in a stronger privacy guarantee. The DP guarantee places an information-theoretic limit on an adversary's ability to infer whether a specific individual is present in the dataset.

As of today, DP has become the de facto standard for defining privacy and we have seen several real-world deployment of DP at Google[Erlingsson et al., 2014], Apple [Apple, Differential Privacy Team, 2017] and U.S. Census Bureau [Rodriguez et al.]. DP is now undergoing an exciting transition from theory to practice, and the goal of my research is to make it more practical to the extent that it can truly solve real-life privacy problems. In this thesis, we explore two directions to enhance DP mechanisms with better privacy and utility trade-offs: modern privacy accountings and data-adaptive analysis. Privacy accountings aims at characterizing the privacy degration over compositions, which is a fundamental topic in dfferential privacy. Classic privacy accountings does not tightly handle composition or privacy amplification by sampling, as it relies solely on a single pair (ϵ, δ)-DP to characterize each DP mechanism. The key idea behind recent modern privacy accountings is the use of a function to precisely describe the privacy guarantee of each DP mechanism, leading to substantially tighter bounds. In this thesis, we delve into two modern privacy accountings: Rényi differential privacy and the PLD formalism. We investigate their optimal bounds under composition / amplification by sampling. Second, data-adaptive DP algorithms stand as another classic recipe to improve the utility of DP mechanisms, allowing for less noise injection when the input dataset is *nice*. In this thesis, we generalize two classic DP algorithms ‘propose-test-release" and the "sparse-vector-technique" with data adaptive analysis, which significantly broaden their applications in real-world DP scenarios.

1.1.1 Optimal bounds in modern privacy accountings

Privacy amplification by subsampling under modern privacy accountings framework is the main workhorse behind recent success in differentially private deep learning. In this part, we provide a nearly optimal amplificaiton by (Poisson) sampling bound under the Rényi DP framework. We show that our results directly translate into practical improvements in private deep learning. Secondly, we propose a unification of recent advances in modern privacy accountings (Rényi DP, f-DP and the PLD formalism) via the *characteristic function*. We show that our approach allows natural adaptive composition as Rényi DP, provides *exactly tight* privacy accounting like PLD. This part is adapted from the paper titled "Poisson Subsampled Rényi Differential Privacy" [Zhu and Wang, 2019] and the paper titled “Optimal Accounting of Differential Privacy via Characteristic Function” [Zhu et al., 2022].

1.1.2 Modern privacy accountings with private deep learning

In Part 2, we focus on the problem of improving private deep learning with better algorithms. We propose Private k-Nearest Neighbor (Private-kNN), the first practical differentially private deep learning solution for large-scale computer vision that achieves theoretically meaningful DP guarantees ($\epsilon < 1$). Our approach allows the use of privacy-amplification by subsampling and iterative refinement of the kNN feature embedding. Moreover, we revisit the problem of private predcition and propose a new algorithm named Individual Kernelized Nearest Neighbor (Ind-KNN). Ind-KNN is easily updatable over dataset changes and it allows precise control of the Renyi DP at an individual user level — a user’s privacy loss is measured by the exact amount of her contribution to predictions; and a user is removed if her prescribed privacy budget runs out. Our results show that Ind-KNN consistently improves the accuracy over existing private prediction methods for a wide range of ϵ on four vision and language task. This part is adapted from the paper titled “Private-knn: Practical Differential Privacy for Computer Vision” [Zhu et al., 2020] and the paper titled “Privacy Prediction Strikes Back!” Private Kernelized Nearest Neighbors with Individual Rényi Filter” [Zhu et al., 2023].

1.1.3 Making classic DP mechanisms practical via data-adaptive analysis

In Part 3, we focus on improving the classic DP algorithms with better utilities via data-adaptive analysis. The “Propose-Test-Release” (PTR) is a powerful tool for deisigning data-adaptive DP algorithms, but applies only to noise-adding mechanisms.

We extend PTR to a more general setting by privately testing data-dependent privacy losses rather than local sensitivity, hence making it applicable beyond the standard noise-adding mechanisms. Our results also broaden the applicability of private hyperparameter tuning in enabling joint selection of DP specific parameters (e.g., noise level) and native parameters of the algorithm (e.g., regularization). The Sparse Vector Technique (SVT) is a foundamental tool in differential privacy (DP) that allows the algorithm to screen potentially an unbounded number of adaptively chosen queries while paying a cost of privacy only for a small number of queries that passes a predefined threshold. We revisit SVT from the lens of Rényi differential privacy, which results in new privacy bounds, new theoretical insight and new variants of SVT algorithm with better utilies.

This part is adapted from the paper titled "Improving Sparse Vector Technique with Rényi Differential Privacy" [Zhu and Wang, 2020] and the paper titled "Generalized PTR: User-Friendly Receipes for Data-Adaptive Algorithms with Differential Privacy" [Redberg et al., 2023].

1.2 Background on Differential Privacy

This section imtroduces the definition of differential privacy (DP) and the relevant DP mechanisms that will be used throught this thesis. We begin with the formal definition of differential privacy, then describe a few important properties of DP. We then detail selected DP mechanisms, which will bed served as building blocks in this thesis.

1.2.1 Defining Differential Privacy

Differential privacy (DP) is a quantifiable and composable definition of privacy that provides provable guarantees against identifications of individuals in a data set. It imposes constraints on the extent to which a single data point can influence the resulting output. We define two dataset D and D' are neighboring if they differ in at most one entry.

Definition 1.2.1 (Differential privacy[Dwork et al., 2006])**.** A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private (DP) if for any pair of neighboring dataset D and D', and any $O \subset \mathrm{Range}(\mathcal{M})$,

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in O] + \delta.$$

. The definition places an information-theoretic limit on an adversary's ability to infer whether the input dataset is D or D', and as a result, guarantees a degree of *plausible deniability* to any individual in the population. ϵ, δ are privacy loss parameters that quantify the strength of privacy protection. In practice, we consider the privacy guarantee marginally meaningful if $\epsilon \approx 1$ and $\delta = o(1/n)$[1], where n denotes the size of data set and $o(\cdot)$ is the standard little-o notation. When $\delta = 0$, we say that $\mathcal{M}$ obeys ϵ-(pure) DP.

One important property of DP relevant to this thesis is that it composes gracefully over multiple access. Roughly speaking, if we run k sequentially chosen (ϵ, δ)-DP algorithm on a dataset, the overall *composed* privacy loss is $(\tilde{O}(\sqrt{k}\epsilon), k\delta + \delta')$-DP where the $\tilde{O}$ notation hides logarithmic terms in k, $1/\delta$ and $1/\delta'$.

[1] It is traditionally required that δ to be cryptographically small, e.g., $o(\mathrm{poly}(1/n))$, but in practice, with a big data set, $\delta = 1/n^2$ is typically considered acceptable.

1.3 An exmple of modern privacy accounting: Rényi Differential Privacy

Renyi Differential Privacy and Moments Accountant. Renyi differential privacy (RDP) is a refinement of DP that uses Renyi-divergence as a distance metric in the place of the sup-divergence.

Definition 1.3.1 (Rényi Differential Privacy [Mironov, 2017])**.** We say that a mechanism $\mathcal{M}$ is (α, ϵ)-RDP with order $\alpha \in (1, \infty)$ if for all neighboring datasets X, X'

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

Through this thesis, we do not treat each α in isolation but instead take a functional view of RDP where we use $\epsilon_{\mathcal{M}}(\alpha)$ to denote that randomized algorithm $\mathcal{M}$ obeys $(\alpha, \epsilon_{\mathcal{M}}(\alpha))$-RDP. The function $\epsilon_{\mathcal{M}}(\cdot)$ can be viewed as a more elaborate description of the privacy loss incurred by running $\mathcal{M}$. It subsumes pure-DP as an RDP algorithm is $\epsilon(+\infty)$-DP.

The moments accountant technique [Abadi et al., 2016] can be thought of as a data structure that keeps track of the RDP (function) for the sequence of data accesses. Composition is trivial in RDP as

$$\epsilon_{\mathcal{M}_1\times\mathcal{M}_2}(\cdot) = [\epsilon_{\mathcal{M}_1} + \epsilon_{\mathcal{M}_2}](\cdot).$$

At any given time, let the composition of all algorithms being $\mathcal{M}$, the moments accountant can be used to produce an (ϵ, δ)-DP certificate using

$$\delta \Rightarrow \epsilon : \qquad \epsilon(\delta) = \min_{\alpha > 1} \frac{\log(1/\delta)}{\alpha - 1} + \epsilon_{\mathcal{M}}(\alpha - 1), \tag{1.1}$$

$$\epsilon \Rightarrow \delta : \qquad \delta(\epsilon) = \min_{\alpha > 1} e^{(\alpha-1)(\epsilon_{\mathcal{M}}(\alpha-1)-\epsilon)}. \tag{1.2}$$

This approach is simpler and often produces more favorable composed privacy parameters than the advanced composition approach for (ϵ, δ)-DP. As the moments accountant gain popularity, many classes of randomized algorithms with exact analytical RDP are becoming available, e.g., the exponential family mechanisms [Geumlek et al., 2017].

(ϵ, α)-RDP implies $(\epsilon(\alpha) + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP, thus by viewing RDP as a function $\epsilon_{\mathcal{M}}(\cdot)$, we can find the best ϵ parameter by optimizing over α. Tighter conversion formula had been proposed recently [Balle et al., 2020, Asoodeh et al., 2021].

Part I

Optimal Bounds in Modern Privacy Accountings

Chapter 2

Privacy-amplification by (Poisson) subsampling with Rényi DP

2.1 Introduction

“Privacy-amplification by Subsampling” and the Renyi Differential Privacy are the two fundamental techniques that have been driving many exciting recent advances in differentially private learning [Abadi et al., 2016, Park et al., 2016, Papernot et al., 2018, McMahan et al., 2018].

One prominent use case of both techniques is the NoisySGD algorithm [Song et al., 2013, Bassily et al., 2014, Wang et al., 2015, Foulds et al., 2016, Abadi et al., 2016] for differentially private deep learning. NoisySGD iteratively updates the model parameters as follows:

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \left(\sum_{i \in \mathcal{I}} \nabla f_i(\theta_t) + Z_t \right) \tag{2.1}$$

where θ_t is the model parameter at tth step, η_t is the learning rate, f_i is the loss function of data point i, ∇ is the standard gradient operator, $\mathcal{I} \subset [n]$ is a randomly subsampled index set and $Z_t \sim \mathcal{N}(0, \sigma^2 I)$. When $\nabla f_i(\theta_t)$ is bounded (or clipped) in ℓ_2-norm, the Gaussian noise-adding procedure is known to ensure (ϵ, δ)-DP for this iteration. ϵ, δ are nonnegative numbers that quantifies the privacy loss incurred from running the algorithm (the smaller the better). But this is clearly not good enough as it takes many iterations to learn the model, and the privacy guarantee deteriorates as the algorithm continues. This is where the "privacy-amplification" and RDP become useful.

The principle of "privacy-amplification by subsampling" works seamlessly with NoisySGD as it allows us to exploit the randomness in choosing the minibatch $\mathcal{I}$ for the interest of a stronger privacy guarantee. Roughly speaking, if the minibatch $\mathcal{I}$ is obtained by selecting each data point with probability γ, then we can "amplify" the privacy guarantee to a stronger $(O(\gamma\epsilon), \gamma\delta)$-DP.

The RDP framework provides a complementary set of benefits that reduce the overall privacy loss over the multiple iterations we run NoisySGD. Notice that the vanilla "strong-composition" is stated for any (ϵ, δ)-DP algorithm. By using the moments accountant techniques [Abadi et al., 2016] that keep track of the RDP of a specific algorithm — subsampled-Gaussian mechanism, one can hope to more efficiently use the privacy budget than what an optimal algorithm would be able to using only (ϵ, δ)-DP [Kairouz et al., 2015].

In general, however, calculating the RDP for the procedure that first subsamples the data set then apply a randomized mechanism $\mathcal{M}$ is highly non-trivial. An exact analytical formula is not known even for the widely-used subsampled-Gaussian mechanism. Existing asymptotic bounds are typically off by a constant, and only apply to a restricted subset of the parameter regimes. To get the most mileage out of the moments accountant, practitioners often resort to numerical integration which calculates and keep track of a fixed list of RDP values [Abadi et al., 2016, Park et al., 2016].

Wang et al. [2019b] took a first stab at this problem and provided a general "RDP-amplification" bound that applies to any $\mathcal{M}$. Their result, however, is still a constant factor away from being optimal. A more subtle difference is that Wang et al. [2019b] considered "Subsampling without Replacements" — finding a random subset of size m at random — rather than the "Poisson subsampling" that was used by Abadi et al. [2016], which includes each data points independently at random with probability γ. The difference is substantial enough that it introduces several new technical hurdles.

In this work, we provide the first general result of "privacy-amplification" of RDP via Poisson subsampling. Our main contributions are the following.

1. First, we prove a nearly optimal upper bound on the RDP of $\mathcal{M} \circ \mathsf{PoissonSample}$ as a function of the sampling probability γ, RDP order α, and the RDP of $\mathcal{M}$ up to α. The bound matches a lower bound up to an additive factor of $\log(3)/(\alpha-1)$, where α is the order of RDP. When α is small relative to $1/\gamma$ with γ being the sampling probability, our upper bound is optimal up to a multiplicative factor of $1+O(\gamma\alpha e^{\epsilon(\alpha)})$. The result tightens and generalizes Lemma 3 of [Abadi et al., 2016], which addresses only the case when $\mathcal{M}$ is Gaussian mechanism and applies only to the cases when γ is very small.

2. Second, we identify a novel condition on the odd order Pearson-Vajda χ^α-Divergences under which we can exactly attain the lower bound. We show that Gaussian mechanism and Laplace mechanism fall under this category, but there exists $\mathcal{M}$ that samples from an exponential family distribution where the condition is false and the lower bound is not attainable. Practically, our analytical characterization simplifies the moments accountant approach for differentially private deep learning by avoiding numerical integration and pre-specifying a list of moments. On the theory front, our result corroborates the observation of Wang et al. [2019b] that the

Pearson-Vajda Divergences are natural quantities for understanding the subsampling in differential privacy.

3. Lastly, knowing that exactly evaluating the analytical subsampled RDP bound of αth order takes α calls of the RDP subroutine $\epsilon_{\mathcal{M}}(\cdot)$, we propose an efficiently τ-term approximation scheme that uses only τ call of $\epsilon_{\mathcal{M}}(\cdot)$. We conduct numerical experiments to compare our general bounds, tight bound, and τ-term approximations for a variety of problem setup and showcasing the use of these bounds in moments accountant-based strong composition.

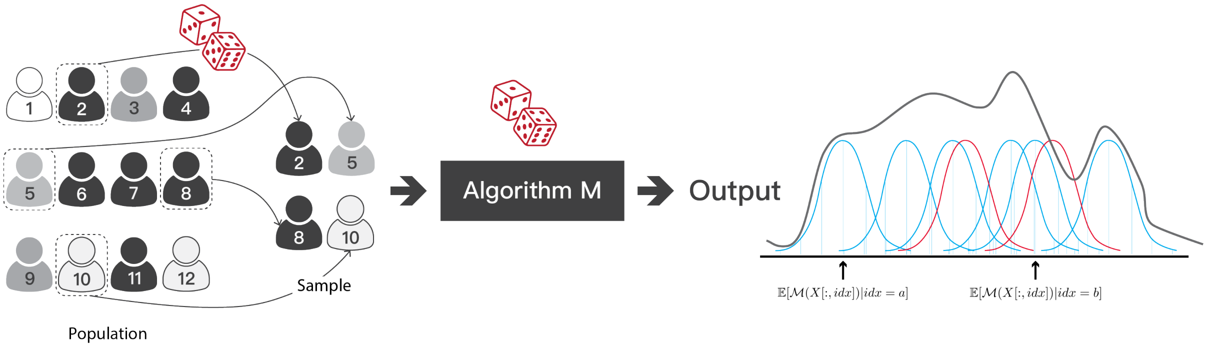

1
2
3
4
5
6
7
8
9
10
11
12
2
5
8
10
Sample
Population
Algorithm M
Output
$\mathbb{E}[\mathcal{M}(X[:, idx])|idx = a]$
$\mathbb{E}[\mathcal{M}(X[:, idx])|idx = b]$

Figure 2.1: Illustration of the subsampled-mechanism and the key underlying idea that enables "privacy-amplification". The diagram on the left illustrate the two parts of randomization. Part (1): PoissonSample: Each person toss a random coin to select whether they are included in the data set; Part (2): The subsampled data set is analyzed by a randomized algorithm $\mathcal{M}$. The figure on the right illustrates the fact that the distribution of output is a mixture distribution indexed by the different potential subset selected by the subsampling, and that when we change the original data set by adding or removing one person, only a small fraction of the mixture components that *happen to be affected* by that change will be different, hus opening up the possibility of "privacy amplifying".

2.2 Background and Problem Setup

In this section, we provide some background on differential privacy, privacy-amplification by subsampling, RDP and the moments accountant technique so as to formally set up the problem. We will also introduce symbols and notations as we proceed.

One important property of DP relevant to this paper is that it composes gracefully over multiple access. Roughly speaking, if we run k sequentially chosen (ϵ, δ)-DP algorithm on a dataset, the overall *composed* privacy loss is $(\tilde{O}(\sqrt{k}\epsilon), k\delta + \delta')$-DP where the $\tilde{O}$ notation hides logarithmic terms in k, $1/\delta$ and $1/\delta'$. Part of the reason for writing this paper is to enable sharper algorithm-dependent composition for a popular class of algorithms that subsamples the data first. Before we get there, let us describe the RDP framework and the moments accountant that the make these algorithm-dependent composition possible.

Moments Accountant.

As a side note, the initial moments accountant [Abadi et al., 2016] keeps track of a vector of log-moment (equivalent to RDP up to a rescaling) associated with a predefined list of order αs. Wang et al. [2019b] observes that these optimization problems are unimodal and proposes an *analytical moments accountant* that solves (1.1) and (1.2) using bisections can be solved using bisection with a doubling trick. This avoids the need to pre-define the list of moments to track. Wang et al. [2019b] also observes that $(\alpha - 1)\epsilon(\alpha)$ is a convex function in α and any such discretization scheme (e.g., all integer α) can be extended into a continuous function in α by simply doing linear interpolation.

Privacy amplification by subsampling. As we discussed in the introduction, "privacy amplification by subsampling" is the other workhorse (besides RDP / moments accountant) that drove much of the recent advances in differentially private deep learning. We would like to add that, it was also used as a key technical hammer for analyzing DP algorithms for empirical risk minimization [Bassily et al., 2014] and Bayesian learning [Wang et al., 2015], as well as for studying learning-theoretic questions with differential privacy constraints [Kasiviswanathan et al., 2011, Beimel et al., 2013, Bun et al., 2015, Wang et al., 2016].

We now furnish a bit more details on this central property and highlight some subtleties in the types. The privacy amplification lemma was derived in [Kasiviswanathan et al., 2011, Beimel et al., 2013, Li et al., 2012], where all three authors adopted what Balle et al. [2018] calls Poisson subsampling:

Definition 2.2.1 (PoissonSample)**.** Given a dataset X, the procedure PoissonSample outputs a subset of the data $\{x_i|\sigma_i = 1, i \in [n]\}$ by sampling $\sigma_i \sim \mathrm{Ber}(\gamma)$ independently for $i = 1, ..., n$.

The procedure is equivalent to the "sampling without replacement" scheme with $m \sim \mathrm{Binomial}(\gamma, n)$. At the limit of $n \to \infty$, $\gamma \to 0$ while $\gamma n \to \lambda$, the Binomial distribution converges to a Poisson distribution with parameter λ. This is probably the reason why it is called Poisson sampling to begin with[1].

Here we cite the tight privacy amplification bound for PoissonSample as it first appears.

Lemma 2.2.2 ([Li et al., 2012, Theorem 1])**.** *If $\mathcal{M}$ is (ϵ, δ)-DP, then $\mathcal{M}'$ that applies $\mathcal{M} \circ$ PoissonSample obeys (ϵ', δ')-DP with $\epsilon' = \log\big(1 + \gamma(e^\epsilon - 1)\big)$ and $\delta' = \gamma\delta$.*

The lemma implies that if the base privacy loss $\epsilon \leq 1$, then the amplified privacy loss obeys that $\epsilon' \leq 2\gamma\epsilon$.

Poisson subsampling is different from the "sampling without replacement" scheme that outputs a subset with size γn uniformly at random. Interestingly, it was shown that the latter also enjoys the same bound with respect to the "replace-one" version of the DP definition. In general, we find that the "add/remove" version of the DP definition works more naturally with Poisson sampling, while the "replace-one" version works well

[1] We noticed that the original definition of Poisson sampling in the survey sampling theory is slightly more general. It allows a different probability of sampling each person [Särndal et al., 2003]. Our results apply trivially to that setting as well with a personalized RDP bound for individual i that depends on γ_i.

with “sampling without replacement”. We defer a more comprehensive account of the subsampling lemma for (ϵ, δ)-DP to [Balle and Wang, 2018] and the references therein.

Subsampled RDP and friends. A small body of recent work focuses on deriving algorithm-specific subsampling Lemma so that this classical wisdom can be combined with more modern techniques such as RDP and Concentrated DIfferential privacy (CDP) [Bun and Steinke, 2016] (also [Dwork and Rothblum, 2016]). Abadi et al. [2016] obtains the first such results for subsampled-Gaussian mechanism under Poisson subsampling. Wang et al. [2019b] provides a general subsampled RDP bound that supports any $\mathcal{M}$ but under the “sampling without replacement” scheme. The objective of this paper is to come up with results of a similar flavor for the Poisson sampling scheme. The main differences in our setting include:

(a) Poisson sampling goes naturally with add/remove version of the DP definition, which is independent to the size of the data.

(b) The size of the random subset m itself is a Binomial random variable.

(c) It is asymmetric, the Renyi divergence of P against Q is different from the Renyi divergence of Q against P.

As we will see in the our results, the third difference brings about some major technical challenges.

Finally, Bun et al. [2018] studies subsampling in CDP with a conclusion that subsampling does not amplify the CDP parameters in general. A truncated version of CDP was then proposed, called tCDP, which does get amplified up to a threshold. CDP and tCDP are closely related to RDP in that they are linear upper bounds of $\epsilon(\alpha)$ on $(1, \infty]$ and on $(1, \tau]$ for some threshold τ respectively. RDP captures finer information about the underlying mechanism. The experimental results in [Wang et al., 2019b] suggest that unlike the case for the Gaussian mechanism (in which case CDP is tight), there isn’t a good linear approximation of $\epsilon(\alpha)$ for the subsampled-Gaussian mechanism due to the phase transition. Our results on the Poisson-sampling model echoes the same phenomenon.

More symbols and notations. We end the section with a quick summary of the notations that we introduced. X, X' denotes two neighboring datasets. $\mathcal{M}$ is a randomized algorithm and $\epsilon_{\mathcal{M}}(\cdot)$ is the RDP function of $\mathcal{M}$ (the subscript may be dropped when it's clear from the context). n, m are reserved for the size of the original and subsampled data. We note that neither is public and m is random. Greek letters $\alpha, \gamma, \epsilon, \delta$ are reserved for the order of RDP, the sampling probability as well as the two privacy loss parameters. $\mathcal{M} \circ \mathsf{PoissonSample}(X)$ is used to mean the composition function $\mathcal{M}(\mathsf{PoissonSample}(X))$.

Let us also define a few shorthands. We will denote p to be the density function of $\mathcal{M} \circ \mathsf{PoissonSample}(X)$, and q to be the density from data set $\mathcal{M} \circ \mathsf{PoissonSample}(X')$. Similarly, we will define μ_0 and μ_1 as two generic density functions of $\mathcal{M}(X)$ and $\mathcal{M}(X')$.

2.3 Main results

Before we present our main result, we would like to warn the readers that the presented bounds might not be as interpretable. We argue that this is a feature rather than an artifact of our proof because we need the messiness to state the bound exactly. These bounds are meant to be *implemented* to achieve the tightest possible privacy composition numerically in the Moments Accountant, rather than being made easily interpretable. After all, "constant matters in differential privacy!" For the interest of interpretability, we provide figures that demonstrate the behaviors of the bound for prototypical mechanisms in practice.

Theorem 2.3.1 (General upper bound)**.** *Let $\mathcal{M}$ be any randomized algorithm that obeys*

$(\alpha, \epsilon(\alpha))$-RDP. Let γ be the subsampling probability and then we have for integer $\alpha \geq 2$,

$$\epsilon_{\mathcal{M}\circ\mathsf{PoissonSample}}(\alpha) \leq \frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1) + \binom{\alpha}{2}\gamma^2(1-\gamma)^{\alpha-2}e^{\epsilon(2)} + 3\sum_{\ell=3}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\ell)}\Big\}.$$

The proof is revealing but technically involved. One main difference from Wang et al. [2019b] is that in Poisson sampling we need to bound both $D_\alpha(p\|q)$ and $D_\alpha(q\|p)$. Existing arguments via the quasi-convexity of Renyi divergence allows us to easily bound $D_\alpha(p\|q)$ tightly using RDP for the case when p has one more data points than q, but $D_\alpha(q\|p)$ turns out to be very tricky. A big part of our novelty in the proof is about analyzing $D_\alpha(q\|p)$. We defer more details of the proof to d 2.5.1.

Theorem 2.3.2 (Lower bound)**.** *$\mathcal{M}$ and pairs of adjacent data sets such that*

$$\epsilon_{\mathcal{M}\circ\mathsf{PoissonSample}}(\alpha) \geq \frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1) + \sum_{\ell=2}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\ell)}\Big\}.$$

Proof: The construction effectively follows Proposition 11 of [Wang et al., 2019b], while adjusting for the details. Let $\mathcal{M}$ be Laplace noise adding of a counting query $f(X') = \sum_{x\in X'}\mathbf{1}[x > 0]$. Let everyone in the data set X' obeys that $x < 0$. In the adjacent dataset $X' = X \cup \{x_{n+1}\}$ with $x_{n+1} > 0$. Let μ_0 be the Laplace distribution centered at 0, μ_1 be the one that is centered at 1. Then we know that $\mathcal{M}(X') \sim \mu_0 = q$ and $\mathcal{M}(X)$ $(1-\gamma)\mu_0 + \gamma\mu_1 = p$. It follows that

$$\mathbb{E}_q[(p/q)^\alpha] = \mathbb{E}_{\mu_0}[((1-\gamma)+\gamma\mu_1/\mu_0)^\alpha] = \sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}\mathbb{E}_{\mu_0}[(\mu_1/\mu_0)^{\ell}].$$

By definition $\mathbb{E}_{\mu_0}[(\mu_1/\mu_0)^\ell] = e^{(\ell-1)D_\ell(\mu_0\|\mu_1)}$. which the RDP bonud $\epsilon(\ell)$ is attained by μ_0, μ_1, then we have constructed one pair of p, q, which implies a lower bound for RDP of $\mathcal{M} \circ \mathsf{PoissonSample}$. ■

Note that the only difference between the upper and lower bounds are a factor of 3 on the third summand in side the logarithm. In the regime when $\gamma\alpha e^{\epsilon(\alpha)} \ll 1$ (in which case the third summand is much smaller than the second), the upper and lower bound match up to a multiplicative factor of $1 + O(\gamma\alpha e^{\epsilon(\alpha)})$. In all other regimes, the upper and lower bounds match up to an additive factor of $\frac{\log(3)}{\alpha-1}$. The results suggest that we can construct a nearly optimal moment accountant.

Remark 2.3.3 (Nearly optimal Moment Accountant). This implies that if *any algorithm* with the help of an oracle that calculates the exact RDP for $\mathcal{M}$ is able to prove an (ϵ, δ)-DP for the Poisson subsampled RDP mechanism, then the RDP upper bound we construct using Theorem 2.3.1 will lead to an $(\epsilon, 3\delta)$-DP bound for the same mechanism.

Moreover, we show that for many randomized algorithms (including the popular Gaussian mechanism and Laplace mechanism) that satisfy an additional assumption, we can strengthen the upper bound further and *exactly match* the lower bound for all α.

Theorem 2.3.4 (Tight upper bound)**.** *Let $\mathcal{M}$ be a randomized algorithm with up to αth order RDP $\epsilon(\alpha) < \infty$. If for all adjacent data sets $X \sim X'$, and all odd $3 \leq \ell \leq \alpha$,*

$$D_{\chi^\ell}(\mathcal{M}(X)\|\mathcal{M}(X')) := \mathbb{E}_{\mathcal{M}(X')}\left(\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1\right)^\ell \geq 0, \tag{2.2}$$

then the lower bound in Theorem 2.3.2 is also an upper bound.

The proof of this theorem is presented in Appendix 2.5.1

In the theorem, $\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1$ is a linearized version of the privacy random variable $\log\frac{\mathcal{M}(X)}{\mathcal{M}(X')}$ and $D_{\chi^\ell}(\mathcal{M}(X)\|\mathcal{M}(X'))$ is the Pearson-Vajda χ^ℓ pseudo-divergence [Vajda, 1973], which has more recently been used to approximate any f-divergence in [Nielsen and Nock, 2014]. The related $|\chi^\ell|$ version of this divergence is identified as the key quantity *natural* for studying subsampling without replacement [Wang et al., 2019a].

The non-negativity condition requires, roughly speaking, the distribution of the linearized privacy loss random variable $\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1$ to be skewed to the right.

The following Lemma provides one way to think about it.

Lemma 2.3.5. *Let π, μ be two measures that are absolute continuous w.r.t. each other and let $\alpha \geq 1$.*

$$\mathbb{E}_\mu[(\pi/\mu - 1)^\alpha] = \mathbb{E}_\pi[(\pi/\mu - 1)^{\alpha-1}] - \mathbb{E}_\mu[(\pi/\mu - 1)^{\alpha-1}].$$

Proof: $\mathbb{E}_\mu[(\pi/\mu - 1)^\alpha] = \mathbb{E}_\mu[(\pi/\mu - 1)(\pi/\mu - 1)^{\alpha-1}] = \mathbb{E}_\pi[(\pi/\mu - 1)^{\alpha-1}] - \mathbb{E}_\mu[(\pi/\mu - 1)^{\alpha-1}]$ ■

The lemma implies that (2.2) holds if an only if for all even $2 \leq \ell \leq \alpha$

$$\mathbb{E}_{\mathcal{M}(X)}\left(\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1\right)^\ell \geq \mathbb{E}_{\mathcal{M}(X')}\left(\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1\right)^\ell$$

for all pairs of X, X'.

This should intuitively be true for most mechanisms because we know from nonnegativity that $\frac{\mathcal{M}(X)}{\mathcal{M}(X')} - 1 \geq -1$, which poses a hard limit to which you can be skewed to the left. Indeed, we can show that the condition is true for the two most widely used DP procedure.

Proposition 2.3.6. *Condition* (2.2) *is true when $\mathcal{M}$ is the Gaussian mechanism and Laplace mechanism.*

The proof, given in Appendix 2.5.2, is interesting and can be used as recipes to qualify other mechanisms for the tight bound. The main difficulty of checking this condition is

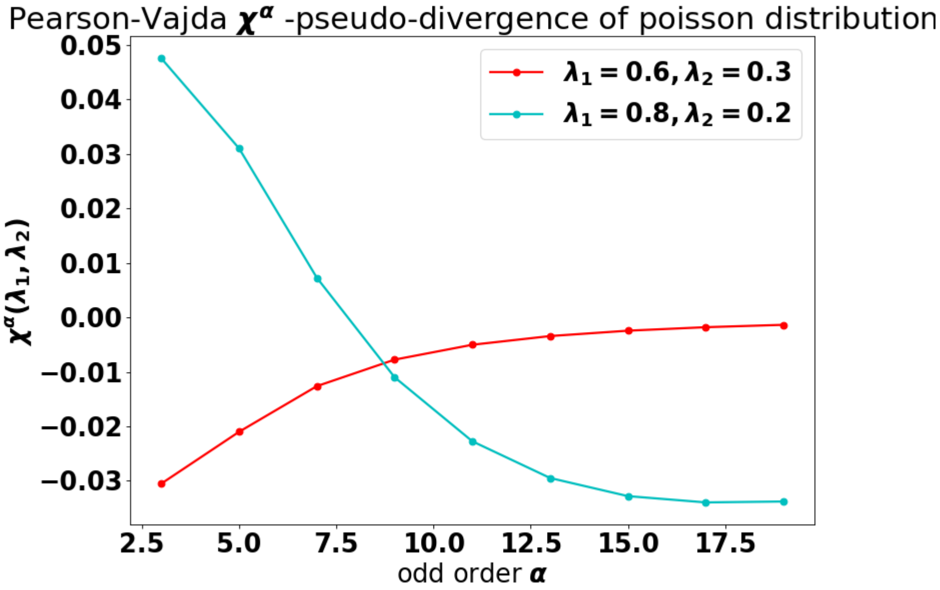


Figure 2.2: Negative χ^α divergence in Poisson distribution.

in searching all pairs of neighboring data sets and identify one pair that minimizes the odd order moment. The convenient property of noise-adding procedure is that typically the search reduces to a univariate optimization problem of the sensitivity parameter.

One may ask, whether the condition is true in general for any randomized algorithm $\mathcal{M}$? The answer is unfortunately no. For example, Nielsen and Nock [2014] constructed an example of two Poisson distributions with negative χ^3-divergence (see Figure 2.2 for an illustration.) This also implies that for some $\mathcal{M}$, we can derive a lower bound that is greater than that in Theorem 2.3.2 by simply toggling the order of X and X'. As a result, if one needs to work out the tight bound, the condition needs to be checked for each $\mathcal{M}$ separately.

Finally, we address the computational issue of implementing our bounds in moments accountant. Naive implementation of Theorem 2.3.1 will easily suffer from overflow or underflow and it takes α calls to the RDP oracle $\epsilon_{\mathcal{M}}(\cdot)$ before we can evaluate one RDP of the subsampled mechanism at α. This is highly undesirable. The following theorem provides a fast approximation bound that can be evaluated with just 2τ calls to the RDP oracle of $\mathcal{M}$. The idea is that since either it is the first few terms that dominates the sum or the last few terms that dominates the sum, we can just compute them exactly and calculate the remainder terms with a more easily computable upper bound.

Theorem 2.3.7 (τ-term approximation)**.** *The expression in Theorem 2.3.2 (therefore Theorem 2.3.4) can be bounded by*

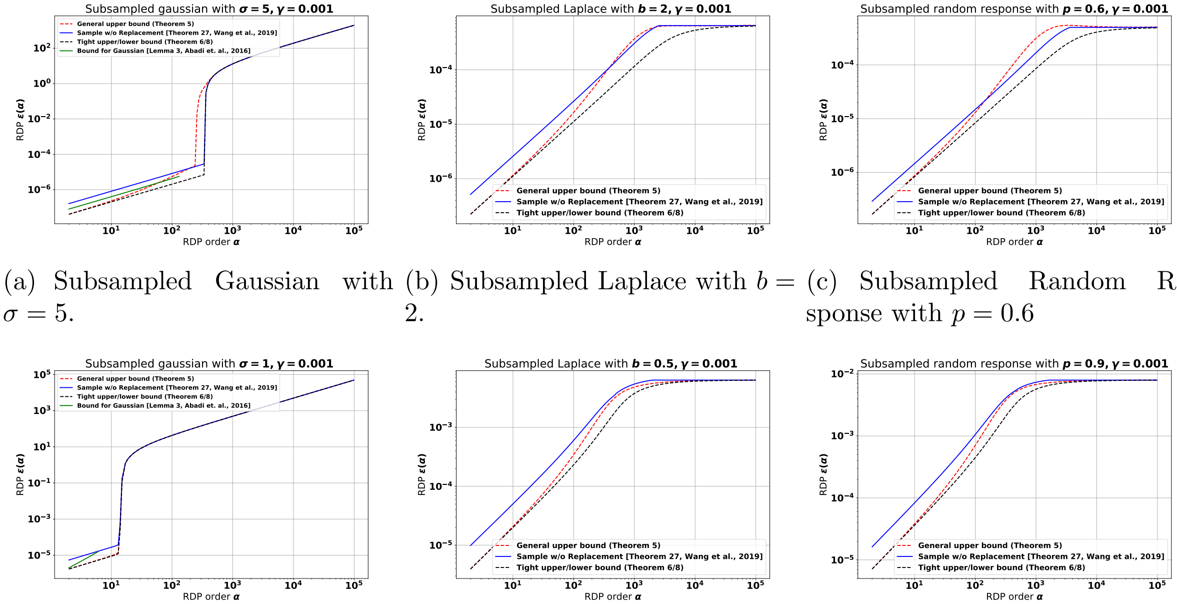
$$\begin{aligned}&\epsilon_{\mathcal{M}\circ\mathsf{PoissonSample}}(\alpha) \leq \frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha}(1-e^{-\epsilon(\alpha-\tau)})\\&+e^{-\epsilon(\alpha-\tau)}(1-\gamma+\gamma e^{\epsilon(\alpha-\tau)})^{\alpha}\\&-\sum_{\ell=2}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}(e^{(\ell-1)\epsilon(\alpha-\tau)}-e^{(\ell-1)\epsilon(\ell)})\\&+\sum_{\ell=\alpha-\tau+1}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}(e^{(\ell-1)\epsilon(\ell)}-e^{(\ell-1)\epsilon(\alpha-\tau)})\Big\}.\end{aligned}$$

A similar bound can be stated for the general upper bound in Theorem 2.3.1, which we defer to Appendix 2.5.3.

Remark 2.3.8 (Numerical stability)*.* The bounds in Theorem 2.3.1 and 2.3.4 can be written as the $\log-\mathrm{sum}-\exp$ form, i.e., softmax. The numerically stable way of evaluating $\log-\mathrm{sum}-\exp$ is well-known. The bound in Theorem 2.3.7, though, have both positive terms and negative terms. We choose to represent the summands in the log term by a sign, and the logarithm of its magnitude. This makes it possible for us to use $\log-\mathrm{diff}-\exp$ and compute the bound in a numerically stable way.

2.4 Experiments and Discussion

In this section, we conduct various numerical experiments to illustrate the behaviors of the RDP for subsampled mechanisms and showcasing its usage in moments accountant



(a) Subsampled Gaussian with $\sigma = 5$.

(b) Subsampled Laplace with $b = 2$.

(c) Subsampled Random Response with $p = 0.6$

(d) Subsampled Gaussian with $\sigma = 1$

(e) Subsampled Laplace with $b = 0.5$.

(f) Subsampled Random Response with $p = 0.9$

Figure 2.3: The RDP parameter ($\epsilon(\alpha)$) of three subsampled mechanisms as a function of order α. The subsampling rate $\gamma = 0.001$ in all the experiments. The upper six figures demonstrate the comparison of general upper bound with other methods under high and low privcy regime. Note that moment $\sigma = 5, b = 2, p = 0.6$. For Approximate RDP upper bound obtained through Theorem 2.3.7, and the corresponding tight upper bound in possion subsample case is represented as the black curve.

for composition. We will have three set of experiments. (1) We will just plot our RDP bounds (Theorem 2.3.1, Theorem 2.3.2) as a function of α. (2) We will compare how close the τ-term approximations approximate the actual bound. (3) We will build our moments accountant and illustrate the stronger composition that we get out of our tight bound.

Specifically, for each of the experiments above, we replicate the experimental setup of which takes the base mechanism $\mathcal{M}$ to be Gaussian mechanism, Laplace mechanism and Randomized Response mechanism. Their RDP formula are worked out analytically

[Mironov, 2017] below:

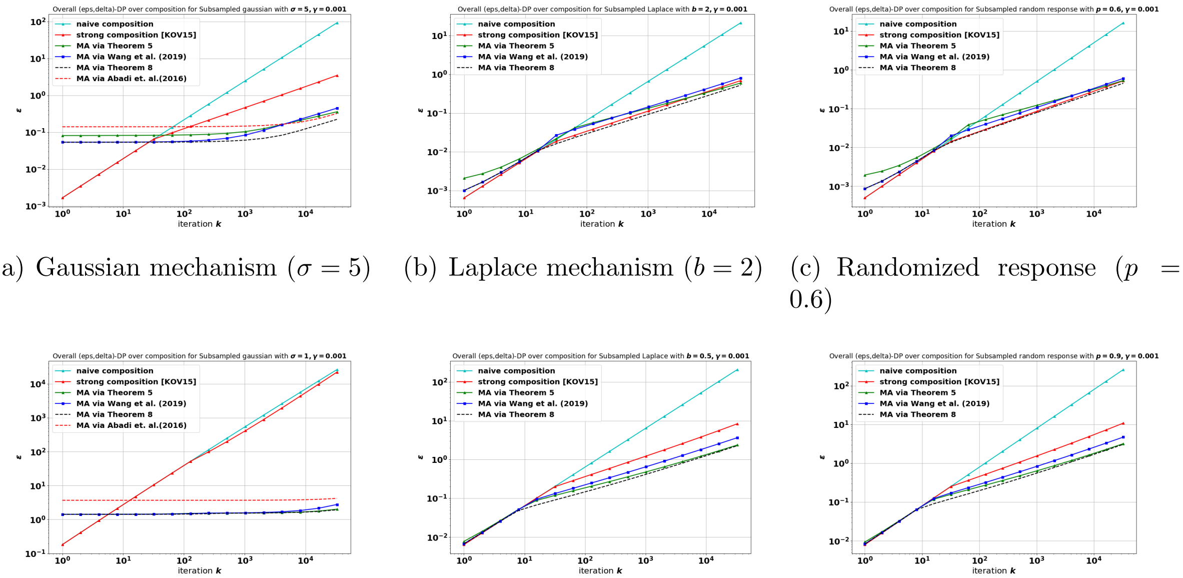
$$
\begin{aligned}
\epsilon_{Gaussian(\alpha)} &= \frac{\alpha}{2\sigma^2}, \\
\epsilon_{Laplace(\alpha)} &= \frac{1}{\alpha-1}\log((\frac{\alpha}{2\alpha-1})e^{\frac{\alpha-1}{b}} + (\frac{\alpha-1}{2\alpha-1})e^{\frac{-\alpha}{b}} \text{ for } \alpha > 1, \\
\epsilon_{RandResp(\alpha)} &= \frac{1}{\alpha-1}\log(p^{\alpha}(1-p)^{1-\alpha} + (1-p)^{\alpha}p^{1-\alpha}) \text{ for } \alpha > 1,
\end{aligned}
$$

where parameter σ, b, p are the standard parameters for Gaussian, Laplace and Bernoulli distributions.

Following Wang et al. [2019a], we will have two sets of experiments with "high noise, high privacy" setting $\sigma = 5, b = 2$, and $p = 0.6$ and "low noise, low privacy" setting using $\sigma = 1, b = 0.5, p = 0.9$. These parameters are chosen such that the ϵ-DP or (ϵ, δ)-DP of the base mechanisms are roughly $\epsilon \approx 0.5$ in the high privacy setting or $\epsilon \approx 2$ in the low privacy setting.

We will include benchmarks when appropriate. For example, we will compare to Lemma 3 of Abadi et al. [2016] whenever we work with Gaussian mechanisms. Also, we will compare to the upper bound of Wang et al. [2019a] for subsample without replacements. Finally, we will include the more traditional approaches of tracking and composing privacy losses using simply (ϵ, δ)-DP. We will see that while the moments accountant approach does not dominate the traditional approach, it does substantially reduces the aggregate privacy loss for almost all experiments when we compose over a large number of rounds.

Comparing RDP bounds. The results on the RDP bounds are shown in the first two rows of Figure 2.3. First of all, the RDP of subsampled Gaussian mechanism behaves very differently from that of the Laplace mechanism, There is a phase transition about the subsampled-Gaussian mechanism that happens around $\alpha\gamma e^{\epsilon(\alpha)} \approx \gamma^{-1}$. Before the phase transition the RDP is roughly $O(\gamma^2\alpha^2(e^{\epsilon(2)} - 1))$, the RDP quickly converges to $\epsilon(\alpha)$, which implies that subsampling has no effects. This kind of behaviors cannot be captured



(a) Gaussian mechanism ($\sigma = 5$) (b) Laplace mechanism ($b = 2$) (c) Randomized response ($p = 0.6$)

(d) Gaussian mechanism ($\sigma = 1$) (e) Laplace mechanism ($b = 0.5$) (f) Randomized response ($p = 0.9$)

Figure 2.4: Illustration of the use of our bounds in moments accountant. We plot the the privacy loss ϵ for $\delta = 1e - 8$ (using (1.1)) after k rounds of composition. The x-axis is the number of composition k,and the y-axis is the privacy loss after k's composition. The green curve is based on general upper bound for all parametrized random mechanism obtained through Theorem 2.3.1. Short hand MA refer to "moment accountant". The upper three figures are in high privacy regime with parameter $\sigma = 5, b = 2, p = 0.6$, the lower three are in low privacy regime with $\sigma = 1, b = 0.5, p = 0.9$.

through CDP. On the other hand, for ϵ-DP mechanisms, the RDP increases linearly with α before being capped what the standard privacy amplification by Lemma 2.2.2. Relative to existing bounds, our tight bound closes the constant gap, while our general bound is also nearly optimal as we predict. It is worth noting that the bound in Abadi et al. [2016] only applies to up to a threshold of α.

τ-term approximation. The third and fourth row illustrates the quality of approximation as we increase τ. With $\tau = 50$, the results nearly matches the RDP bound everywhere, except that in the Gaussian case, the phase-transition happened a little bit earlier.

Usage in moments accountant. The experiments on moments accountant are shown in Figure 2.4. Our result are compared to the optimal strong composition [Kairouz et al., 2015] with parameters optimally tuned according to Wang et al. [2019a]. As we can see, all bounds based on the moments accountants eventually scales proportional to $\sqrt{k}$. Moments accountant techniques with the tight bound end up winning by a constant factor. It is worth noting that in the Gaussian case, moments accountant only starts to perform better than traditional approaches after composing for 1000 times. Also, the version of moments accountant using the theoretical bounds from Abadi et al. [2016] gave substantially worse results[2]. Finally the general RDP bound perform as well as the tight bound when k is large (thanks to the tightness for small α).

2.5 Conclusion

In this work, we study the problem of privacy-amplification by poisson subsampling, which involves "add/remove" scheme instead of replacement strategy. Specifically, we derive a tight upper bound for $\mathcal{M} \circ \text{PoissonSample}$ for any mechanism satisfying that their odd order Pearson-Vajda χ^α-Divergences are nonnegative. We showed that Gaussian mechanism and Laplace mechanism have this property, as a result, finding the exact analytical expression for the Poisson subsampled Gaussian mechanism that has seen significant application in differentially private deep learning. Our results imply that we can completely avoid numerical integration in moments accounts and track the entire range of α without paying unbounded memory. In addition, we propose an efficiently τ-term approximation scheme which only calculates the first and last τ terms in the Binomial expansion when evaluating the RDP of subsampled mechanisms. This greatly simplifies

[2] We implemented the bound from the proof of Abadi et al. [2016]'s Lemma 3 for fair comparison. According to Section 3.2 of Abadi et al. [2016], their experiments use numerical integration to approximate the moments. See more discussion on this in Appendix 2.5.4.

the computation for computing ϵ given δ as is used in the moments accountant. The experiment result of τ-term approximate part reveals that approximate bound matches up the lower bound quickly even for a relative small τ.

Future work includes making use of the exact subsampled RDP bounds to tighten the existing results that made use of subsampled-mechanisms, coming up with more general recipe to automatically check the nonnegativity condition on the odd-order Pearson-Vajda χ^α-Divergences and design differentially private learning algorithms with more complex and hetereogenous building blocks.

2.5.1 Proof of Theorem 2.3.1 and Theorem 2.3.4

Recall that we will denote the density of $\mathcal{M} \circ \mathsf{PoissonSample}(X')$ by q and that of $\mathcal{M} \circ \mathsf{PoissonSample}(X)$ by p. Let's first make a few observations.

1. There is a natural change of measure that we can do:

$$\mathbb{E}_q e^{\alpha \log(p/q)} = \mathbb{E}_q[(p/q)^\alpha] = \mathbb{E}_p[(p/q)^{\alpha-1}] = \mathbb{E}_p[e^{(\alpha-1)\log(p/q)}].$$

   This relates RDP to the moment generating function of the log-odds ratio random variable, or the privacy random variable $\log(p/q)$.

2. With our loss of generality, we can assume $X' = X \cup \{x\}$. In order to bound RDP with order α, it suffices to bound the moments $\mathbb{E}_p[(q/p)^\alpha]$ and $\mathbb{E}_q[(p/q)^\alpha]$ then take the bigger of the two bounds.

3. Both p and q are mixture distributions. Let $|X| = n-1$ and $|X'| = n$. p has 2^{n-1} mixture components and q has 2^n mixture components. Each component corresponds to a particular subset of the data set.

4. If we condition on condition on $J = (\sigma_1, ..., \sigma_{n-1}) \in \{0,1\}^{n-1}$, we get

$$\mathbb{E}_p[(q/p)^\alpha] = \int \frac{\left(\sum_J \mathbb{P}(J)\left[(1-\gamma)\mu_0(J) + \gamma\mu_1(J)\right]\right)^\alpha}{(\sum_J \mathbb{P}(J)\mu_0(J))^{\alpha-1}}$$

By Lemma 23 of [Wang et al., 2019b], $f(x,y) = x^\alpha/y^{\alpha-1}$ is jointly convex on $\mathbb{R}^2_+$ for all $\alpha \in (1,+\infty)$, which allows us to apply Jensen's inequality to get

$$\mathbb{E}_p[(q/p)^\alpha] \leq \sum_J \mathbb{P}(J)\mathbb{E}_{\mu_0(J)}\left(\frac{(1-\gamma)\mu_0(J) + \gamma\mu_1(J)}{\mu_0(J)}\right)^\alpha$$

and similarly

$$\mathbb{E}_q[(p/q)^\alpha] \leq \sum_J \mathbb{P}(J)\mathbb{E}_{(1-\gamma)\mu_0(J)+\gamma\mu_1(J)}\left(\frac{\mu_0(J)}{(1-\gamma)\mu_0(J) + \gamma\mu_1(J)}\right)^\alpha.$$

where μ_0 is the distribution of $\mathcal{M}(X_J)$ and μ_1 is the distribution of $\mathcal{M}(X_j \cup \{x\})$. [3]

Denote $\mu_0 := \mu_0(J)$ and $\mu_1 := \mu_1(J)$ as short hands. What matters is that μ_0 and μ_1 are distributions induced by the application of our base mechanism $\mathcal{M}$ to two adjacent data sets.

The fourth observation reduces the problem to bounding $A_1 := \mathbb{E}_{\mu_0}\left[\left(\frac{(1-\gamma)\mu_0+\gamma\mu_1}{\mu_0}\right)^\alpha\right]$ and $A_2 := \mathbb{E}_{(1-\gamma)\mu_0+\gamma\mu_1}\left[\left(\frac{\mu_0}{(1-\gamma)\mu_0+\gamma\mu_1}\right)^\alpha\right]$ using RDP of $\mathcal{M}$.

Let's start with A_1 and consider only the case when $\alpha \geq 1$ is an integer. Also, without loss of generality, we assume $\gamma < 1$. Let $\alpha \geq 1$ be an integer, and assume $\gamma < 1$. By the

[3] Note that the arguments used by Abadi et al. [2016] based on the quasi-convexity of Renyi-divergence will give a slightly weaker result but with the expectation over J replaced with the maximum over J, which will be sufficient for our purpose too in this paper.

Binomial theorem:

$$\begin{aligned}
A_1 =& \mathbb{E}_{\mu_0}\left[\left((1-\gamma)+\gamma\frac{\mu_1}{\mu_0}\right)^\alpha\right] \\
=& \sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell \\
=& (1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1)+\sum_{\ell=2}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell \\
\leq& (1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1)+\sum_{\ell=2}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell e^{(\ell-1)\epsilon(\ell)}.
\end{aligned}$$

A_2 is tricky as we cannot always calculate it explicitly or approximate efficiently with Renyi-divergence. By a change of measure, we can write A_2 as moments of a negative order.

$$A_2 = \mathbb{E}_{\mu_0}\left[\left(1-\gamma+\gamma\frac{\mu_1}{\mu_0}\right)^{-(\alpha-1)}\right].$$

Trivially, we have two somewhat trivial upper bounds

$$A_2 \leq (1-\gamma)^{-(\alpha-1)}. \tag{2.3}$$

When $\mathcal{M}$ is ϵ-DP,

$$A_2 \leq (1-\gamma(1-e^{-\epsilon}))^{-(\alpha-1)}.$$

Other than these two, the expression does not seem to give us a more meaningful bound. It might be tempted to use Binomial series expansion (now an infinite series). However, it is not guaranteed to converge for some μ_0, μ_1. Even in cases when it converges, we will have positive and negative terms that we could not construct a tight expression with RDP.

A novel alternative decomposition.

Let us try to bound $\mathbb{E}_q[(p/q)^\alpha]$ through an alternative means. We will redefine the index set $J = (\sigma_1, ..., \sigma_n) \subset \{0,1\}^n$.

Define $q' = \sum_J \mathbb{P}(J) q'(J)$ such that $q'(J) = q((\sigma_1, ..., \sigma_{n-1}, 1))$. Define $p' = \sum_J \mathbb{P}(J) p'(J)$ such that $p'(J) = q((\sigma_1, ..., \sigma_{n-1}, 0))$. Note that $p = p'$, $q = (1-\gamma)p + \gamma q'$ and therefore $p = q + \gamma p' - \gamma q'$.

It follows from Jensen's inequality and the joint convexity that

$$\begin{aligned}
\mathbb{E}_q[(p/q)^\alpha] =& \mathbb{E}_q\left[\left(\frac{q + \gamma p' - \gamma q'}{q}\right)^\alpha\right] \leq \mathbb{E}_J \mathbb{E}_q\left[\left(\frac{q + \gamma p' - \gamma q'}{q}\right)^\alpha\right] \\
\leq& \mathbb{E}_{\sigma_1,...,\sigma_{n-1}} \mathbb{E}_{\sigma_n} \mathbb{E}_{q(J)}\left[\left(\frac{q(J) + \gamma p'(J) - \gamma q'(J)}{q(J)}\right)^\alpha\right] \\
=& \mathbb{E}_{\sigma_1,...,\sigma_{n-1}} \Big\{ \gamma \mathbb{E}_{q'(J)}\left[\left(\frac{q'(J) + \gamma p'(J) - \gamma q'(J)}{q'(J)}\right)^\alpha \middle| \sigma_n = 1\right] \\
&+ (1-\gamma)\mathbb{E}_{p'(J)}\left[\left(\frac{p'(J) + \gamma p'(J) - \gamma q'(J)}{p'(J)}\right)^\alpha \middle| \sigma_n = 0\right] \Big\} \\
=& \gamma \mathbb{E}_{\mu_1}\left[\left(\frac{(1-\gamma)\mu_1 + \gamma\mu_0}{\mu_1}\right)^\alpha\right] + (1-\gamma)\mathbb{E}_{\mu_0}\left[\left(\frac{(1+\gamma)\mu_0 - \gamma\mu_1}{\mu_0}\right)^\alpha\right] \qquad (2.4) \\
=& \gamma \mathbb{E}_{\mu_1}\left[\left((1-\gamma) + \gamma\frac{\mu_0}{\mu_1}\right)^\alpha\right] + (1-\gamma)\mathbb{E}_{\mu_0}\left[\left(1 - \gamma + \gamma(2 - \frac{\mu_1}{\mu_0})\right)^\alpha\right] \\
=& \sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell}\gamma^\ell \left\{ \gamma \mathbb{E}_{\mu_1}\left(\frac{\mu_0}{\mu_1}\right)^\ell + (1-\gamma)\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell \right\} \qquad (2.5)
\end{aligned}$$

There are two interesting things about the above chain of derivation. (2.4) really allows us to evaluate the quantity for any pair of μ_0 and μ_1. However, we cannot really easily upper bound it for all μ_1, μ_2 easily since some of the terms are negative.

Meanwhile, (2.5) is a slightly prettier form. If we can show that $\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell \leq \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell$, then we are done. In fact, for $\ell = 0, 1, 2$, it is straightforward to show that $\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell = \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell$. For $\ell \geq 3$, it becomes quite a deep question whether it is true that $\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell \leq \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell$.

Our first attempt establishes that this is related to the sign of Pearson-Vajda pseudo-divergences of odd orders.

Lemma 2.5.1. *For any pairs of distribution μ_0, μ_1 such that the Renyi-divergence $D_\alpha(\mu_1, \mu_0)$ exists up to order ℓ.*

$$\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell = \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell - 2\sum_{j \textit{ is odd}, j\leq \ell}\binom{\ell}{j}\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0} - 1\right)^j,$$

where $\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0} - 1\right)^j$ is the Pearson-Vajda χ^j-pseudo-divergence of μ_1 and μ_2.

Proof: Observe that $2 - \mu_1/\mu_0 = 1 - (\mu_1/\mu_0 - 1)$ and that $\mu_1/\mu_0 = 1 + (\mu_1/\mu_0 - 1)$. It follows that

$$\begin{aligned}&\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell - \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell \\ &= \sum_{j=0}^{\ell}\binom{\ell}{j}\left\{((-1)^j - 1)\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0} - 1\right)^j\right\} \\ &= -2\sum_{j \text{ is odd}, j\leq\ell}\binom{\ell}{j}\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0} - 1\right)^j\end{aligned}$$

■

Proof: [Proof of Theorem 2.3.4] Note that Condition (2.2) implies

$$\sum_{j \text{ is odd}, j\leq\ell}\binom{\ell}{j}\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0} - 1\right)^j \geq 0 \tag{2.6}$$

as a result, Lemma 2.5.1 implies that

$$\mathbb{E}_{\mu_0}\left(2 - \frac{\mu_1}{\mu_0}\right)^\ell \leq \mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}\right)^\ell \leq e^{(\ell-1)\epsilon_{\mathcal{M}}(\ell)}.$$

Substitute the above into (2.5), then we can obtain a bound identical to the lower bound in the Theorem 2.3.2. ■

A bigger question is that what if the condition above is not true? Can we obtain a general-purpose bound that applies to all $\mathcal{M}$ without needing to worry about whether Condition (2.2) is true.

One idea is to directly evaluate $\sum_{j \text{ is odd}, j\leq \ell} \binom{\ell}{j}\mathbb{E}_{\mu_0}\left(\frac{\mu_1}{\mu_0}-1\right)^j$ and replace all Renyi-divergences of μ_1, μ_0 with the corresponding RDP bound. This is not really a valid argument. We cannot directly evaluate irwith RDP because it is not straightforward how we can take supremum over μ_1, μ_0 (two neighboring data sets). Substituting the RDP into it is not really correct, because there might be some pair of μ_1, μ_0 that do not match the RDP bound.

Can we still obtain a bound that is quantitatively the same as Theorem 2.3.4?

In the following we write two lemmas that allow us to prove such a general purpose bound (Theorem 2.3.1).

Approximation upper bound for $\ell \geq 3$.

Lemma 2.5.2 (Relax order of RDP)**.**

$$\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\right] \leq \begin{cases} e^{\ell\epsilon(\ell+1)} & \textit{if } \ell \textit{ is odd} \\ e^{(\ell-1)\epsilon(\ell)} + e^{\ell\epsilon(\ell+1)} & \textit{if } \ell \textit{ is even.} \end{cases}$$

Proof: We consider decomposing the expression to several pieces.

$$\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\right] = \mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\mathbf{1}(\frac{\mu_1}{\mu_0}\leq 2)\right] \tag{2.7}$$

$$+\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\mathbf{1}(\frac{\mu_1}{\mu_0}> 2)\right] \tag{2.8}$$

In the first term, we use the basic inequality that $\frac{\mu_1}{\mu_0} + \frac{\mu_0}{\mu_1} \geq 2$, which implies that

$$\begin{aligned}0 &\leq \mathbb{E}_{\mu_0}\left[\left(2 - \frac{\mu_1}{\mu_0}\right)^{\ell} \mathbf{1}(\frac{\mu_1}{\mu_0} \leq 2)\right] \leq \mathbb{E}_{\mu_0}\left[\left(\frac{\mu_0}{\mu_1}\right)^{\ell} \mathbf{1}(\frac{\mu_1}{\mu_0} \leq 2)\right] \\ &\leq \mathbb{E}_{\mu_0}\left[\left(\frac{\mu_0}{\mu_1}\right)^{\ell}\right] \leq e^{\ell \epsilon_{\mathcal{M}}(\ell+1)}.\end{aligned}$$

The second term is negative if ℓ is an odd number. Moreover, we can bound its absolute value:

$$\begin{aligned}\left|\mathbb{E}_{\mu_0}\left[\left(2 - \frac{\mu_1}{\mu_0}\right)^{\ell} \mathbf{1}(\frac{\mu_1}{\mu_0} > 2)\right]\right| &= \mathbb{E}_{\mu_0}\left[\left(\frac{\mu_1}{\mu_0} - 2\right)^{\ell} \mathbf{1}(\frac{\mu_1}{\mu_0} > 2)\right] \\ &\leq \mathbb{E}_{\mu_0}\left[\left(\frac{\mu_1}{\mu_0}\right)^{\ell}\right] \leq e^{(\ell-1)\epsilon_{\mathcal{M}}(\ell)}.\end{aligned}$$

In addition, in the case of pure DP with $\epsilon \leq \log(2)$, we have that $\mu_1 \leq e^{\epsilon}\mu_0 \leq 2\mu_0$, which implies that the second term is 0. ■

Lemma 2.5.3 (Relax multiplicative constant)**.**

$$\mathbb{E}_{\mu_0}\left[\left(2 - \frac{\mu_1}{\mu_0}\right)^{\ell}\right] \leq \begin{cases} 2e^{(\ell-1)\epsilon(\ell)} & \textit{if } \ell \textit{ is odd} \\ 3e^{(\ell-1)\epsilon(\ell)} & \textit{if } \ell \textit{ is even.} \end{cases}$$

Proof: We start with the case when ℓ is even.

$$\begin{aligned}
&\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\right] \\
&=\mathbb{E}_{\mu_0}\left[(2-\frac{\mu_1}{\mu_0})\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\right] \\
&=2\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\right]-\mathbb{E}_{\mu_1}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\right] \qquad (2.9)\\
&=2\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\right]+\mathbb{E}_{\mu_1}\left[\left(\frac{\mu_1}{\mu_0}-2\right)^{\ell-1}\right]
\end{aligned}$$

Note that we used the fact that $\ell-1$ is odd in the last line. By Lemma 2.5.2 we can bound the first term by $e^{(\ell-1)\epsilon(\ell)}$. Now by the fact that $x^{\ell-1}$ is a monotonically increasing function, we can bound the second term by $\mathbb{E}_{\mu_1}\left[\left(\frac{\mu_1}{\mu_0}\right)^{\ell-1}\right]$, which also is smaller than $e^{(\ell-1)\epsilon(\ell)}$. That gives us the constant multiplicative factor of 3.

Now consider the case when ℓ is odd. Decompose the expression by (2.8) and drop the second term since it is negative, we can write

$$\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\right]\leq\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell}\mathbf{1}(\frac{\mu_1}{\mu_0}\leq 2)\right].$$

Now apply the same trick as we did to get (2.9), we can rewrite the above as

$$2\mathbb{E}_{\mu_0}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\mathbf{1}(\frac{\mu_1}{\mu_0}\leq 2)\right]-\mathbb{E}_{\mu_1}\left[\left(2-\frac{\mu_1}{\mu_0}\right)^{\ell-1}\mathbf{1}(\frac{\mu_1}{\mu_0}\leq 2)\right].$$

Again note that the second term is negative, and by $\frac{\mu_1}{\mu_0}+\frac{\mu_0}{\mu_1}\geq 2$, we can bound the first term by

$$\begin{aligned}2\mathbb{E}_{\mu_0}\left[\left(\frac{\mu_0}{\mu_1}\right)^{\ell-1}\mathbf{1}(\frac{\mu_1}{\mu_0}\leq 2)\right] &\leq 2\mathbb{E}_{\mu_0}\left[\left(\frac{\mu_0}{\mu_1}\right)^{\ell-1}\right]\\ &= 2\mathbb{E}_{\mu_1}\left[\left(\frac{\mu_0}{\mu_1}\right)^{\ell}\right]\\ &\leq 2e^{(\ell-1)\epsilon(\ell)}.\end{aligned}$$

■

Now we are ready to present the main theorem.

Proof: [Proof of Theorem 2.3.1] Substituting the results in Lemma 2.5.3 to (2.5), relax the constant to 3 and then apply the RDP upper bound of the Renyi-divergence.

■

2.5.2 Tight bounds for Gaussian and Laplace mechanism

In this section, we prove Proposition 2.3.6 and also that our tight bound Theorem 2.3.4 applies to the Gaussian mechanism and Laplace mechanism. In particular, we will show that the condition (2.2) in Theorem 2.3.4 that requires the Pearson-Vajda χ^α divergences to be nonnegative for the π, μ that come running either the Gaussian mechanism or the Laplace mechanism on any two adjacent data sets.

The proof for the Gaussian mechanism uses a novel inductive argument, while the proof for the Laplace mechanism directly proves that moving $f(X')$ away from $f(X)$ strictly increases the odd-order Pearson-Vajda χ^α divergence using tools from convex optimization.

These calculations are possible because the discrepancy of two data sets can be fully described by a single parameter. The general recipe used in this section can also be applied to other cases where only a small number of parameters can be used to avoid the intractable search over any pair of data sets to find the worst pair.

Qualifying Gaussian Mechanism

Lemma 2.5.4. *For any π, μ that are absolutely continuous, and an odd $\alpha \geq 3$,*

$$E_\mu(\frac{\pi}{\mu})^2(\frac{\pi}{\mu} - 1)^{\alpha-2} \geq E_\mu[(\frac{\pi}{\mu} - 1)^{\alpha-2}]$$

Proof:

$$\begin{aligned} E_\mu(\frac{\pi}{\mu})^2(\frac{\pi}{\mu} - 1)^{\alpha-2} &= E_\pi(\frac{\pi}{\mu})(\frac{\pi}{\mu} - 1)^{\alpha-2} \\ &= E_\pi(\frac{\pi}{\mu} - 1)^{\alpha-1} + E_\pi(\frac{\pi}{\mu} - 1)^{\alpha-2} \\ &\geq E_\pi(\frac{\pi}{\mu} - 1)^{\alpha-2} \end{aligned}$$

Since $\alpha - 1$ is even, $E_\mu(\frac{\pi}{\mu} - 1)^{\alpha-1} \geq 0$. $E_\pi(\frac{\pi}{\mu} - 1)^{\alpha-2}$ could be rewritten as $E_\mu(\frac{\pi}{\mu})(\frac{\pi}{\mu} - 1)^{\alpha-2}$

$$\begin{aligned} E_\mu(\frac{\pi}{\mu})(\frac{\pi}{\mu} - 1)^{\alpha-2} &= E_\mu(\frac{\pi}{\mu} - 1)^{\alpha-1} + E_\mu(\frac{\pi}{\mu} - 1)^{\alpha-2} \\ &\geq E_\mu[(\frac{\pi}{\mu} - 1)^{\alpha-2}] \end{aligned}$$

■

Theorem 2.5.5. *Let π, μ be two gaussian distriutions, $\pi \sim \mathcal{N}(\sqrt{t}, 1)$ and $\mu \sim \mathcal{N}(0, 1)$, for $\forall t \geq 0, \forall odd\ \alpha \geq 1$, we have $E_\mu(\frac{\pi}{\mu} - 1)^\alpha \geq 0$.*

Proof: **Base case**: The statement holds when $\alpha = 1$

$$\forall t, E_\mu(\frac{\pi}{\mu} - 1) = 0$$

Inductive step: Show that if $\alpha = \tilde{\alpha}$, we have $E_\mu(\frac{\pi}{\mu}-1)^{\tilde{\alpha}} \geq 0$ for all t, then the statement holds for $\alpha = \tilde{\alpha} + 2$. This can be done as follows:

We first write an expansion of $E_\mu(\frac{\pi}{\mu} - 1)^\alpha$ as :

$$\begin{aligned} E_\mu(\frac{\pi}{\mu} - 1)^\alpha &= \sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (-1)^{\alpha-\ell} E_\mu(\frac{\pi}{\mu})^\ell \\ &= \alpha - 1 + \sum_{\ell=2}^{\alpha} \binom{\alpha}{\ell} (-1)^{\alpha-\ell} e^{\frac{\ell(\ell-1)t}{2}} \end{aligned}$$

When $t = 0$, $E_\mu(\frac{\pi}{\mu} - 1)^\alpha = 0$ holds for all α. We then take the derivative of t on the above expansion.

$$\frac{\partial E_\mu(\frac{\pi}{\mu} - 1)^\alpha}{\partial t} = \sum_{\ell=2}^{\alpha} \binom{\alpha}{\ell} (-1)^{\alpha-\ell} e^{\frac{\ell(\ell-1)t}{2}} \frac{\ell(\ell-1)}{2}$$

Define $\tilde{\ell} = \ell - 2$ and rewrite the above equation as

$$\begin{aligned} &\frac{\alpha(\alpha-1)}{2} \sum_{\tilde{\ell}=0}^{\alpha-2} \binom{\alpha-2}{\tilde{\ell}} (-1)^{\alpha-2-\tilde{\ell}} E_\mu(\frac{\pi}{\mu})^{\tilde{\ell}+2} \\ &= \frac{\alpha(\alpha-1)}{2} E_\mu[(\frac{\pi}{\mu})^2 \sum_{\tilde{\ell=0}}^{\alpha-2} \binom{\alpha-2}{\tilde{\ell}} (-1)^{\alpha-2-\tilde{\ell}} (\frac{\pi}{\mu})^\ell] \\ &= \frac{\alpha(\alpha-1)}{2} E_\mu[(\frac{\pi}{\mu})^2 (\frac{\pi}{\mu} - 1)^{\alpha-2}] \end{aligned}$$

By applying lemma 2.5.4, we have $E_\mu[(\frac{\pi}{\mu})^2(\frac{\pi}{\mu}-1)^{\alpha-2}] \geq E_\mu[(\frac{\pi}{\mu}-1)^{\alpha-2}]$, where $\alpha - 2 = \tilde{\alpha}$ and $E_\mu[(\frac{\pi}{\mu} - 1)^{\tilde{\alpha}}] \geq 0$ from assumption. So the derivative is greater than 0 for all non-negative t. Combined with $E_\mu(\frac{\pi}{\mu} - 1)^\alpha = 0$ when $t = 0$, we have $E_\mu(\frac{\pi}{\mu} - 1)^\alpha \geq 0$ hold for all t.

Since both the base case and the inductive step have been performed, by mathematical induction the statement holds for all odd $\alpha \geq 1$. ■

Qualifying Laplace mechanism

Theorem 2.5.6. *Let π, μ Laplace density functions obeying $\mu(x) = \frac{1}{\lambda}e^{\frac{|x|}{\lambda}}$, and $\pi(x) = \frac{1}{\lambda}e^{\frac{|x+t|}{\lambda}}$. For all $\lambda > 0$, all natural number α, function $f(t) := \mathbb{E}_{\mu}\left[\left(\frac{\pi}{\mu} - 1\right)^{\alpha}\right]$ obeys that*

1. *$f(t) \geq 0$ for any $t \in \mathbb{R}$.*
2. *$f(t)$ monotonically increases for $t > 0$.*
3. *$f(t)$ monotonically decreases for $t < 0$.*

Proof: When $t = 0$, $\pi/\mu = 1$ and trivially $\mathbb{E}_{\mu}[(\pi/\mu - 1)^{\alpha}] = 0$ for any α. We will show that this is actually the minimizer for all $t \in \mathbb{R}$ by proving that the subdifferential $\partial_t f(t) \geq 0$ for $t > 0$ and $\partial_t f(t) \leq 0$ for $t < 0$. Note that

$$\partial_u |u| = \begin{cases} [-1, 1] & \text{if } u = 0; \\ \{\text{sign}(u)\} & \text{otherwise,} \end{cases}$$

which implies that we can write

$$\begin{aligned} \partial_t f(t) = &\int_0^{+\infty} -\frac{\alpha}{2\lambda^2} e^{-\frac{|u|}{\lambda}} \left(e^{\frac{-|u|+|u-t|}{\lambda}} - 1\right)^{\alpha-1} du \\ + &\int_{-\infty}^{0} -\frac{\alpha}{2\lambda^2} e^{-\frac{|u|}{\lambda}} \left(e^{\frac{-|u|+|u-t|}{\lambda}} - 1\right)^{\alpha-1} du \\ = &\int_0^{+\infty} \frac{\alpha}{2\lambda^2} e^{-\frac{|u|}{\lambda}} \left[-\left(e^{\frac{-|u|+|u-t|}{\lambda}} - 1\right)^{\alpha-1} + \left(e^{\frac{-|u|+|u+t|}{\lambda}} - 1\right)^{\alpha-1}\right] du \end{aligned}$$

For positive t, we can decompose the integral into

$$\int_t^{+\infty} \frac{\alpha}{2\lambda^2} e^{-\frac{|u|}{\lambda}} \left[-\left(e^{\frac{-t}{\lambda}} - 1\right)^{\alpha-1} + \left(e^{\frac{t}{\lambda}} - 1\right)^{\alpha-1} \right] du$$
$$+ \int_0^t \frac{\alpha}{2\lambda^2} e^{-\frac{|u|}{\lambda}} \left[-\left(e^{\frac{t-2u}{\lambda}} - 1\right)^{\alpha-1} + \left(e^{\frac{t}{\lambda}} - 1\right)^{\alpha-1} \right] du.$$

For even $\alpha \geq 2$, $\alpha - 1$ is an even number and above expression is trivially nonnegative.

For odd $\alpha \geq 3$, that $\alpha - 1$ is even. By the inequality that $e^t - 1 \geq 1 - e^{-t}$ for any t, therefore the first term is nonnegative.

Now we address the second term. For $u \in [0, t/2]$, $0 \leq t - 2u \leq t$ and the nonnegativity follows directly from the monotonicity of $(e^{\cdot} - 1)$ on $[0, +\infty)$. For $u \in (t/2, t]$, $-t \leq t - 2u \leq 0$, and the nonnegativity follows from the fact that

$$e^t - 1 \geq 1 - e^{-t} \geq 1 - e^{-v}$$

for all $0 \leq v \leq t$. This concludes the proof for the positive t.

The results that the subgradient is positive for negative t follows naturally by symmetry. ■

Remark 2.5.7 (Handling Laplace Mechanism in higher dimension). The generalization to higher dimension is trivial. The perturbation t is now a vector, but since the noise is added independently for each coordinate, we can work out the monotonicity for each coordinate separately.

2.5.3 Proofs related to efficient approximation

Proof: [Proof of Theorem 2.3.7] Apply $\epsilon(\ell) \leq \epsilon(\alpha)$ for all $\ell = \tau+1, ..., \alpha$, we have:

$$\begin{aligned}
&\epsilon_{\mathcal{M}\circ\mathsf{PoissonSample}}(\alpha) \leq \frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1) + \sum_{\ell=2}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\ell)} \\
&+ \sum_{\ell=\tau+1}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\alpha)}\Big\} \\
=&\frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1) + \sum_{\ell=2}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\ell)} \\
&- \sum_{\ell=0}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\epsilon(\alpha)} + e^{-\epsilon(\alpha)}(1-\gamma+\gamma e^{\epsilon(\alpha)})^{\alpha}\Big\} \\
=&\frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha-1}(\alpha\gamma-\gamma+1) + \sum_{\ell=2}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}(e^{(\ell-1)\epsilon(\ell)} - e^{(\ell-1)\epsilon(\alpha)}) \\
&- (1-\gamma)^{\alpha}e^{-\epsilon(\alpha)} - \alpha(1-\gamma)^{\alpha-1}\gamma + e^{-\epsilon(\alpha)}(1-\gamma+\gamma e^{\epsilon(\alpha)})^{\alpha}\Big\} \\
=&\frac{1}{\alpha-1}\log\Big\{(1-\gamma)^{\alpha}(1-e^{-\epsilon(\alpha)}) + e^{-\epsilon(\alpha)}(1-\gamma+\gamma e^{\epsilon(\alpha)})^{\alpha} \\
&-\sum_{\ell=2}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}(e^{(\ell-1)\epsilon(\alpha)} - e^{(\ell-1)\epsilon(\ell)})\Big\}
\end{aligned}$$

■

Theorem 2.5.8 (Fast approximation for general upper bound)**.**

$$\begin{aligned}
\epsilon_{\mathcal{M}\circ\mathsf{PoissonSample}}(\alpha) \leq \frac{1}{\alpha-1}\log\Big\{&(1-\gamma)^{\alpha}(1-3e^{-\epsilon(\alpha)}) + 3e^{-\epsilon(\alpha)}(1-\gamma+\gamma e^{\epsilon(\alpha)})^{\alpha} \\
&- 3\sum_{\ell=3}^{\tau}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}(e^{(\ell-1)\epsilon(\alpha)} - e^{(\ell-1)\epsilon(\ell)}) \\
&- 2\gamma\alpha(1-\gamma)^{\alpha-1} + \binom{\alpha}{2}\gamma^2(1-\gamma)^{\alpha-2}(e^{\epsilon(2)} - 3e^{\epsilon(\alpha)})\Big\}.
\end{aligned}$$

Proof is similar to that of Theorem 2.3.7 thus omitted.

2.5.4 Comparison to the implementation of Abadi et al. [2016]

According to Section 3.2 of Abadi et al. [2016], in the implementation of the moments accountant they used numerical integration to compute

$$E_1 = \mathbb{E}_{z\sim\mu_0}[(\mu_0(z)/\mu(z))^{\alpha-1}] = \mathbb{E}_{z\sim\mu}[(\mu_0(z)/\mu(z))^{\alpha}]$$
$$E_2 = \mathbb{E}_{z\sim\mu}[(\mu(z)/\mu_0(z))^{\alpha-1}] = \mathbb{E}_{z\sim\mu_0}[(\mu(z)/\mu_0(z))^{\alpha}]$$

where $\mu_0 = \mathcal{N}(0,\sigma^2)$ and $\mu = \gamma\mathcal{N}(1,\sigma^2) + (1-\mu)\mathcal{N}(0,\sigma^2)$ then output

$$\epsilon(\alpha) \leq \frac{1}{\alpha-1}\log(\max\{E_1, E_2\}).$$

This approach is correct but costly, because a different numerical integration is needed for each α. Our result implies that $E_2 > E_1$ and one never need to numerically simulate E_1.

The most recent update to the moments accountant implementation of the Tensorflow Privacy package is slightly different from the version described in Section 3.2 of Abadi et al. [2016]. The new version of their code `https://github.com/tensorflow/privacy/blob/master/privacy/analysis/rdp_accountant.py` implements an analytical version of E_2 via the Binomial expansion — essentially our tight bound Theorem 2.3.2 for Poisson-Sampled Gaussian mechanism verbatim with a prescribed list of αs. The current paper complements this implementation with a proof that $E_2 > E_1$, which justifies that doing this is correct. To the best of our knowledge, the current paper is the first that rigorously establishes $E_2 \geq E_1$ which establishes that this new implementation is correct for Poisson subsampling.

Our implementation of moments accountant in AutoDP (`https://github.com/yuxiangw/autodp`) is a more flexible framework that allows us to exactly or almost exactly track the RDP of any subsampled differentially private mechanisms provided that the based mechanism's RDP is known.

Chapter 3

Generalize the PLD formalism with adaptive composition and amplification by sampling

3.1 Introduction

Much of the progress in the recent theory and practice of DP has been driven by Renyi Differential Privacy (RDP) [Mironov, 2017], e.g., it is the major technical component behind the *first practical method* for *deep learning with differential privacy* [Abadi et al., 2016]. More broadly, RDP is among several recent work in differential privacy that conducts fine-grained mechanism specific analysis [Bun and Steinke, 2016, Abadi et al., 2016, Mironov, 2017, Balle and Wang, 2018, Wang et al., 2019b, Dong et al., 2021, Sommer et al., 2019, Koskela et al., 2020a]. At the heart of these breakthroughs is the idea of using a *function* to describe the privacy guarantee of a randomized procedure, thus produces significantly more favorable privacy-utility tradeoff and tighter bounds under composition. (See Table 3.1 for a summary their pros and cons).

	Functional view	Pros	Cons
Renyi DP [Mironov, 2017]	$D_\alpha(P\|Q) \leq \epsilon(\alpha), \forall \alpha \geq 1$	Natural composition	lossy conversion to (ϵ, δ)-DP.
Privacy profile [Balle and Wang, 2018]	$\mathbb{E}_q[(\frac{p}{q} - e^\epsilon)_+] \leq \delta(e^\epsilon), \forall \epsilon \geq 0$	Interpretable.	messy composition.
f-DP[Dong et al., 2021]	Trade-off function f	Interpretable, CLT	messy composition.
PLD [Sommer et al., 2019, Koskela et al., 2020a]	Probability density of $\log(p/q)$	Natural composition via FFT	Limited applicability.

Table 3.1: Modern functional views of DP guarantees and their pros and cons.

Note that no single approach dominates others in all dimensions. Renyi DP could be undefined for certain privacy loss distributions, and cannot be used to provide the optimal (ϵ, δ)-DP computation in general (discussed in Section 3.2). Privacy profiles and f-DP are unwieldy under composition; and the method of [Koskela et al., 2020a] is limited to mechanisms with univariate output where $\log(p/q)$ admits a density; or those with discrete outputs. Usually, for a new mechanism, we would be lucky to have any one of these functional descriptions. The need to derive these manually for each new mechanism is clearly limiting the creativity of researchers and practitioners in DP.

In addition, there are some unresolved foundational issues related to the PLD formalism. As is repeatedly articulated by the authors, the PLD formalism is defined for *each pair* of neighboring datasets separately, thus, strictly speaking, does not imply DP unless we can certify that the pair of neighboring datasets is the worst-case. This is challenging because such a pair of datasets might not exist and it is unclear how we can define a partial ordering of two privacy loss distributions.

In this work, we provide a unified treatment to these functional representations and resolve the aforementioned subtle issues related to the PLD formalism. Our contributions are summarized below.

1. We formalize and generalize the notion of *"worst-case" pair distributions* discussed in [Sommer et al., 2019] to a *"dominating pair"* and prove several basic properties of the *dominating pairs* including finding such pairs from any privacy-profiles, *adaptive* composition and *amplification by sampling.* These results substantially broaden the

applicability of PLD formalism [Sommer et al., 2019] in deriving *worst-case* DP guarantees.

2. We propose a lossless representation of the privacy loss RV by its characteristic function (ϕ-function) and derive optimal conversion formula to (and from) privacy-profile, tradeoff-function (f-DP) and the distribution function of the privacy loss RV. Many of these conversion rules correspond naturally to the classical Fourier / Laplace transforms (and their inverses) from the signal processing literature.

3. We design an Analytical Fourier Accountant (AFA, extending the Fourier accountant of [Koskela et al., 2020a,b]) which represents the complex logarithm of the ϕ function *symbolically.* AFA can be viewed as an extension of the (analytical) moments-accountant [Abadi et al., 2016, Wang et al., 2019b] to complex α, thus allowing straightforward composition. Computing δ as a function of ϵ for (ϵ, δ)-DP boils down to a numerical integral which we use a Gaussian quadrature-based method to solve efficiently and accurately.

4. Experimentally, we demonstrate that our approach provides substantially tighter privacy guarantees over compositions than RDP on both basic mechanisms and their subsampled counterparts. Our results essentially match the results from [Dong et al., 2021] and [Koskela et al., 2020b] but neither rely on central-limit-theorem type asymptotic approximation nor require choosing appropriate discretization a priori as in the FFT-based Fourier Accountant.

Related work: The paper builds upon the existing work on RDP-based privacy accounting [Abadi et al., 2016, Mironov, 2017, Wang et al., 2019b] as well as f-DP [Dong et al., 2021]. Our main theoretical contribution is to substantially broaden the applicability of the PLD formalism [Sommer et al., 2019] by proposing the notion of *dominating* *pairs* and providing *general recipes* for constructing these dominating pairs. The closest to algorithmic contribution is the work of Koskela et al. [2020a,b], who propose Fourier accountant and an FFT-based approximation scheme, the characteristic function view can be seen as an *analytical* version of their Fourier accountant (hence the name AFA). AFA is more generally applicable, and allows more flexible use of existing methods for numerical integral. The recent work of Gopi et al. [2021] improves the FFT accountant substantially. It is complementary to us in that it does not address the foundational issues of the PLD formalism, nor do they propose an analytical representation that allows a more modular design of the privacy accountant. Notably, we can use any blackbox numerical integration tool, e.g., Gaussian quadrature, and set the desired error bound on-the-fly, while an FFT-accountant requires setting the parameters at initialization. Finally, Canonne et al. [2020] considered ϕ function and its numerical / computational properties but the discussion is restricted to the discrete Gaussian mechanism.

Privacy accounting is closely related to the classical advanced composition of (ϵ, δ)-DP [Dwork et al., 2010]; Kairouz et al. [2015] provides the *optimal* k-fold composition of an (ϵ, δ)-DP mechanism and Murtagh and Vadhan [2016] shows that computing the tightest possible bound for the composition of k heterogeneous mechanisms is $\#P$-hard. The recent line of work (that we are building upon) challenges the basic primitive of composing (ϵ_i, δ_i)-DP by composing certain functional descriptions of the mechanisms themselves, which sometimes avoids the computational hardness (but not always) and results in even stronger composition than the best (ϵ, δ)-DP type composition would allow [Bun and Steinke, 2016].

3.1.1 Equivalent definition of DP

We can alternatively interpret DP from the views of a divergence metric of two probability distributions, a hypothesis testing view of a binary-classifier, as well as the distribution of the log-odds ratio. Let us first define these quantities formally.

Definition 3.1.1 (Hockey-stick divergence)**.** For $\alpha > 0$, the Hockey-stick divergence is defined as $H_\alpha(P\|Q) := \mathbb{E}_{o\sim Q}[(\frac{\mathrm{d}P}{\mathrm{d}Q}(o) - \alpha)_+]$, where $(x)_+ := x\mathbf{1}(x \geq 0)$ and $\frac{\mathrm{d}P}{\mathrm{d}Q}$ is the Radon-Nikodym-derivative (or simply the density ratio when density exists for P and Q).

Definition 3.1.2 (Trade-off function)**.** Let ϕ be a classifier to distinguish two distributions P and Q using a sample. α_ϕ be its Type I error (false positive rate) and β_ϕ be its Type II error (false negative rate). The tradeoff function $T_{P,Q}(\alpha) : [0,1] \to [0,1]$ is defined to be $T_{P,Q}(\alpha) := \inf_\phi\{\beta_\phi \mid \alpha_\phi \leq \alpha\}$.

Definition 3.1.3 (Privacy loss R.V.)**.** The *privacy loss random variable* of for a pair of neighboring dataset D, D' under mechanism $\mathcal{M}$ is defined as $L_{P,Q} := \log\frac{\mathcal{M}(D)(o)}{\mathcal{M}(D')(o)}$ where $o \sim \mathcal{M}(D)$; similarly, we have $L_{Q,P} := \log\frac{\mathcal{M}(D')(o)}{\mathcal{M}(D)(o)}$ where $o \sim \mathcal{M}(D')$.

These quantities can be used to equivalently define differential privacy [Wasserman and Zhou, 2010, Barthe and Olmedo, 2013, Kairouz et al., 2015, Balle and Wang, 2018, Balle et al., 2018, Dong et al., 2021].

Lemma 3.1.4. *The following statements about a randomized algorithm $\mathcal{M}$ are equivalent to Definition Definition 1.2.1*

1. $\sup_{D\simeq D'} H_{e^\epsilon}(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \delta$.

2. $\sup_{D\simeq D'} T_{\mathcal{M}(D),\mathcal{M}(D')}(\alpha) \geq \max\{0, 1-\delta-e^\epsilon\alpha, e^{-\epsilon}(1-\delta-\alpha)$.

3. $\Pr_{o\sim\mathcal{M}(D)}[L_{P,Q} > \epsilon] - e^\epsilon \Pr_{o\sim\mathcal{M}(D')}[L_{Q,P} < -\epsilon] \leq \delta$ *for all neighboring D, D'.*

We highlight that in all these definitions, it is required for the bound to cover all pairs of neighboring datasets D, D'.

Mechanism-specific analysis / Functional representation of DP guarantee. Each of these equivalent interpretations could be used to provide more-fine-grained description of a differential privacy mechanism $\mathcal{M}$. For instance, the privacy profile $\delta_{\mathcal{M}}(\epsilon)$ upper bounds the HS-divergence for all ϵ and the f-DP lowerbounds the tradeoff function for all Type I error α (see Table 3.1). In addition, Sommer et al. [2019] proposes the PLD formalism, which represents the privacy loss RV by its density function. The PLD formalism can be viewed as another functional representation, but it is qualitatively different from privacy profile and f-DP. We will expand further on PLD in Section 3.2.

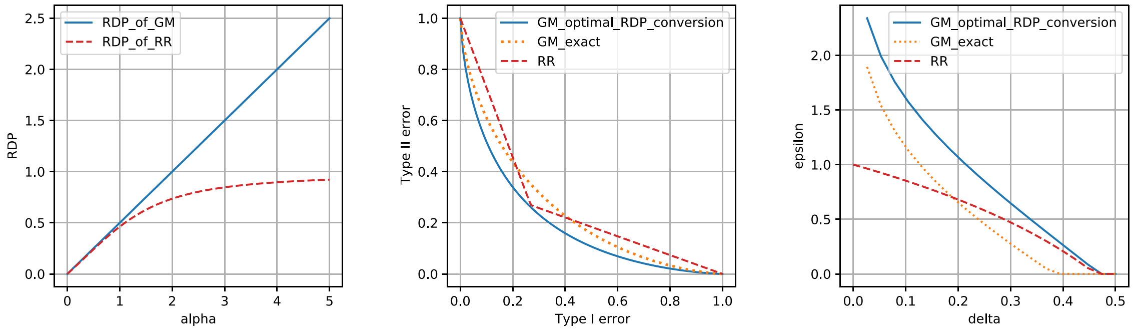
Definition 3.1.5 (Privacy profile [Balle et al., 2018] and f-DP [Dong et al., 2021])**.** The privacy profile of a mechanism $\mathcal{M}$ is a function $\delta_{\mathcal{M}} : \mathbb{R}_{\geqslant 0} \to [0, 1]$defined as

$$\delta_{\mathcal{M}}(\alpha) := \sup_{D \simeq D'} H_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')).$$

$\mathcal{M}$ satisfies f-DP for a tradeoff function $f : [0, 1] \to [0, 1]$ if

$$f(\alpha) \leq f_{\mathcal{M}}(\alpha) =: \inf_{D \simeq D'} T_{\mathcal{M}(D), \mathcal{M}(D')}(\alpha). \tag{3.1}$$

Note that the definition of privacy-profile and the original one in Balle and Wang [2018] differ only in a change of variable $\alpha = e^\varepsilon$. In addition, considering all $\alpha > 0$ amounts to also consider negative ε. Although only $\alpha \geqslant 1$ (or $\varepsilon \geqslant 0$) is involved in the following Lemma 3.1.4 that relates hockey-stick divergence and privacy profile to DP, taking $\alpha \in (0, 1)$ (or $\varepsilon < 0$) into consideration in the above definition is convenient as will be clear in Lemma 3.3.3.



(a) RDP of RR and GM (b) f-DP of RR and GM (c) (ϵ, δ)-DP of RR and GM

Figure 3.1: The figure illustrates the RDP and f-DP of a Gaussian mechanism with (normalized) $\sigma = 1$, and a randomized response mechanism with $p = \frac{e}{1+e}$. Pane (a) shows the RDP function of RR and GM, clearly, RR also satisfies the same RDP of the Gaussian mechanism for all α. Pane (b) in the middle compares the f-DP of the two mechanisms, as well as the f-DP implied by the optimal conversion from RDP. Pane (c) shows the privacy profile of the two mechanisms, together with Pane (a), it demonstrates that the optimal f-DP and (ϵ, δ)-DP of GM cannot be achieved by a conversion from RDP.

3.2 Motivation: limits of RDP and the PLD formalism

In this section, we discuss a number of limitations of Renyi DP and PLD formalism that, in part, motivated our research.

The limits of RDP. Let us first ask "is the RDP function a *lossless* description?" In particular, does it capture all information in the privacy-profile? Because if it is the case, then we could use RDP for composition, and then find the exact optimal (ϵ, δ)-DP by converting from RDP.

The answer is unfortunately "no". The reasons are twofolds. First, there are mechanisms with non-trivial (ϵ, δ)-DP where RDP parameters partially or entirely do not exist. We give two concrete examples below.

Example 3.2.1 (Distance-to-Instability). The stability-based argument of query release first add noise to a special integer-valued function $\mathrm{dist}_q(D)$ which measures the number of data points to add / remove before the local sensitivity of query $q(D)$ becomes non-zero. No matter that q is, dist_q always has a global sensitivity of at most 1. The stability-based query release outputs $\perp$ (nothing) if $\mathrm{dist}_q(D) + \mathrm{Lap}(1/\epsilon) \leq {}^{\log(1/\delta)}/_{\epsilon}$ otherwise outputs the answer $q(D)$ without adding noise. This algorithm is satisfies (ϵ, δ)-DP [Thakurta and Smith, 2013], but since there is a probability mass at the $+\infty$ for the case when $q(D) \neq q(D')$, RDP is $+\infty$ for all α.

Example 3.2.2 (Gaussian-noise adding with data-dependent variance)*.* In smooth sensitivity-based query release [Nissim et al., 2007], one perturbs the output with a noise with a data-dependent variance. Consider, for example, $P = \mathcal{N}(0, \sigma_1^2), Q = \mathcal{N}(0, \sigma_2^2)$, then the Renyi-divergence $D_\alpha(P\|Q)$ is undefined for all α such that $\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2 < 0$. Specifically, if $\sigma_1^2 = 2, \sigma_2^2 = 1$, then $D_\alpha(P\|Q) = +\infty$ for all $\alpha \geq 2$.

These examples demonstrate the deficiency of RDP in analyzing flexible algorithm design tools such as the proposed-test-release [Dwork and Lei, 2009]), which typically introduces a heavier-tailed privacy-loss distributions for which the moment generating function is not defined.

On the contrary, the privacy-profile is well-defined in both examples and imply non-trivial (ϵ, δ)-DP. The characteristic function exists no matter how heavy-tailed the distribution of the privacy loss random variable is so it naturally handles the second example.

The second, and a more troubling issue is that even in the cases when RDP parameters exist everywhere and hence *appears to be* characterizing, it does not lead to a tight conversion to (ϵ, δ)-DP. Gaussian mechanism is such a candidate where its PLD is completely captured by its Renyi divergences. However, in Figure Figure 3.1 we demonstrate that we cannot, in general, convert the RDP of Gaussian mechanism into an (ϵ, δ)-DP that matches the optimal accounting one can achieve through either the privacy profile or f-DP directly. Specifically, by an example due to [Dong et al., 2021, Proposition B.7], we know that a randomized response mechanism (RR) satisfies 1-zCDP, thus the same RDP as that of a Gaussian mechanism (GM) with $\sigma = 1$. If the RDP conversion is tight, then it will have to apply to RR too, but that will lead to a contradiction with the tradeoff function of RR. More explicitly, when we further convert the f-DP in Figure Figure 3.1 to (ϵ, δ)-DP, this example shows that while both RR and GM satisfy an RDP with $\epsilon(\alpha) = \frac{\alpha}{2}$, GM obeys $(0.277, 0.3)$-DP but RR does not satisfy $(\epsilon, 0.3)$-DP with $\epsilon < 0.471$.

This example certifies that the conversion rule we used (based on an extension of [Balle et al., 2020]) *cannot be improved* and that *RDP is a lossy representation* even for the Gaussian mechanism.

Trouble with Worst-Cases in the PLD formalism.

Recent developments in the PLD formalism show great promises in computing tight (ϵ, δ)-DP with stable numerical algorithms and provable error bounds [Koskela et al., 2020a,b]. However, as we discussed earlier, PLD is specified for each pair of input datasets separately. To use PLD, the original authors (quoting verbatim) *"require the privacy analyst interested in applying our results (PLD formalism) to provide worst-case distributions."* [Sommer et al., 2019, Section 2]. In a subsequent work [Meiser and Mohammadi, 2018], a subset of the authors further derive the worst-case pair of distributions for basic mechanisms such as Gaussian mechanism and Laplace mechanism [Meiser and Mohammadi, 2018].

While these are valid arguments, the line of work on PLD formalism does not formally define the worst-case pair of distributions, nor do they provide general recipes for "privacy analysts" to determine which pair of inputs is the worst-case. The issue is more prominent when we consider mechanism-specific analysis, because the pairs of datasets that attain the argmax might be different in different regions of the privacy profile.

Example 3.2.3 (Distance to Instability). Distance to instability $\text{dist}_q(D)$ is a special function that measures the number of data points to add / remove before the local sensitivity of query $q(D)$ becomes non-zero. The stability-based query release outputs $\perp$ (nothing) if $\text{dist}_q(D) + \text{Lap}(1/\epsilon) \leq {}^{\log(1/\delta)}/_{\epsilon}$ otherwise outputs the answer $q(D)$ without adding noise. In this algorithm, the privacy loss distribution has exactly two modes.

Mode 1 When $\text{dist}_q(D) > 0$, then for all D' neighboring to D, $q(D) = q(D')$, which implies that the PLD is from the post-processing of a Laplace mechanism (for releasing the perturbed $\text{dist}_q(D)$), i.e., $(\epsilon, 0)$-DP.

Mode 2 When $\text{dist}_q(D) = 0$, then for those neighboring D' such that $q(D) \neq q(D')$, it must hold that $\text{dist}_q(D') = 0$, thus the privacy loss distribution is a point mass of $1 - \delta$ at 0 (for outputting $\perp$) and a point mass of δ at $+\infty$, i.e., $(0, \delta)$-DP.

Clearly, there is no single pair of datasets that attains the privacy-profile of this mechanism for all input parameter $\tilde{\epsilon}$. When $\tilde{\epsilon} > \epsilon$, $\delta_{\mathcal{M}}(\tilde{\epsilon}) = \delta$ and is attained by the second mode. On the other hand, if we choose $\tilde{\epsilon}$ such that $\delta_{\text{Lap. Mech.}(1/\epsilon)}(\tilde{\epsilon}) > \delta$, then $\delta_{\mathcal{M}}(\tilde{\epsilon}) = \delta_{\text{Lap. Mech.}(1/\epsilon)}(\tilde{\epsilon})$ and the equal sign is attained by a pair of distributions in the first mode.

Moreover, in most typical use cases of the privacy accounting tools, the mechanism under consideration is constructed through the composition of a sequence of simpler mechanisms. Even if for each mechanism, we know the worst-case pair distributions, the composition of the individual PLDs may not correspond to the worst-case PLD of the composed mechanism [1]. For this reason, it is unclear how to use PLD for deriving worst-case DP bound under composition except in highly specialized cases (e.g., Gaussian mechanisms and their compositions).

[1] This is an issue we will address later, which shows that it is OK even if it does not.

Summary. To reiterate, RDP is lossy when converting to (ϵ, δ)-DP and the PLD formalism cannot be used to handle the composition generically due to issues regarding worst-case distributions. The remainder of the paper will be dedicated to addressing this dilemma.

3.3 Main results

In this section, we develop a comprehensive solution towards tighter and more flexible *mechanism-specific* privacy accounting for (ϵ, δ)-DP with a data-structure that allows natural composition.

3.3.1 Dominating pair of distributions, composition and subsampling

We first patch the PLD formalism by generalizing the idea of worst-case pair (which may not exist) to a *dominating* pair of distributions and prove a number of useful properties.

Definition 3.3.1 (Dominating pair of distributions)**.** We say that (P, Q) is a *dominating* pair of distributions for $\mathcal{M}$ (under neighboring relation $\simeq$) if for all $\alpha \geq 0$[2]

$$\sup_{D \simeq D'} H_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq H_\alpha(P \| Q). \tag{3.2}$$

When P, Q is chosen such that (3.2) takes "=" for all α, we say that (P, Q) is a *tight* dominating pair of distributions or simply, *tightly dominating.* If in addition, there exists a neighboring $(\tilde{D}, \tilde{D}')$ such that $(\mathcal{M}(\tilde{D}), \mathcal{M}(\tilde{D}'))$ is *tightly dominating*, and then we say

[2] Note that $\alpha \geq 1$ corresponds to the typical range of (ϵ, δ)-DP, but the region for $\alpha < 1$ is important for composition and lossless conversions to other representations.

$(\tilde{D}, \tilde{D}')$ is the worst-case pair of datasets for mechanism $\mathcal{M}$.

Unless otherwise specified, all subsequent results we present hold for any definitions of neighbors (including asymmetric ones such as *add-only* and *remove-only*, which will be useful later).

A dominating pair of distributions always exists: one can trivially take P and Q that have disjoint supports. What is somewhat surprising is the following

Proposition 3.3.2. *Any mechanism has a* tightly *dominating pair of distributions.*

For example, the domintating pair for discrete Gaussian mechanism (DGM) [Canonne et al., 2020] will be two discrete Gaussian, e.g., $P = \mathcal{NZ}(0, \sigma^2), Q = \mathcal{NZ}(\Delta, \sigma^2), \Delta \in \mathbb{Z}_+$ is the sensitivity of the integer-valued query. This follows because the probability mass of the discrete Gaussian is a log-concave sequence. The proof would look very similar to Proposition A.3 of Dong et al. [2021]. On the other hand, worst-case pair of datasets do not always exist, as is shown by Example 3.2.3.

Proposition 3.3.2 is the direct consequence of the following result which fully characterizes what hockey-stick divergences and privacy profiles look like.

Lemma 3.3.3. *For a given $H : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$, there exists P, Q such that $H(\alpha) = H_\alpha(P\|Q)$ if and only if $H \in \mathcal{H}$ where*

$$\mathcal{H} := \left\{ H : \mathbb{R}_{\geqslant 0} \to \mathbb{R} \,\middle|\, \begin{array}{l} H \text{ is convex, decreasing,} \\ \\ H(0) = 1 \text{ and } H(x) \geqslant (1-x)_+ \end{array} \right\}.$$

Moreover, one can explicitly construct such P and Q: P has CDF $1 + H^*(x-1)$ in $[0,1)$ and $Q = \mathrm{Uniform}([0,1])$.

The proof, presented in Sec 3.5.1, makes use of the Fenchel duality of the privacy profile with respect to a tradeoff function and a characterization of the tradeoff function due to Dong et al. [2021, Proposition 2.2].

What makes the specific construction in Lemma 3.3.3 (hence Proposition 3.3.2) appealing is that even if the output space is complex, the resulting dominating pair of distributions are of univariate random variables defined on $[0, 1]$. This resolves a limitation of Koskela et al. [2020a] that requires the mechanism to have either univariate or discrete outputs.

So far, we have shown the existence of a tightly dominating pairs for all mechanisms (Proposition 3.3.2), and provided a recipe for constructing such a dominating pair for any valid upper bounds of the privacy profile (Lemma 3.3.3 and Corollary 3.5.1 in Sec 3.5.1). Next we will provide two general primitives on how to construct dominating pairs for more complex mechanisms created by composition and privacy amplification by sampling.

Theorem 3.3.4 (Adaptive composition of dominating pairs)**.** *If* (P, Q) *dominates* $\mathcal{M}$ *and* (P', Q') *dominates* $\mathcal{M}'$[3]*, then* $(P \times P', Q \times Q')$ *dominates the composed mechanism* $(\mathcal{M}, \mathcal{M}')$*.*

By induction, this theorem implies that if we construct the PLD using a dominating pair of distributions for each individual mechanism, then the composed PLD can be used to obtain a *valid* worst-case DP of the composed mechanism.

Next we present how we can construct a dominating pair of distributions (and datasets) for mechanisms under "privacy-amplification by sampling". This is a powerful primitive that is used widely in differentially private ERM [Bassily et al., 2014], Bayesian learning [Wang et al., 2015] and deep learning [Abadi et al., 2016]. We consider the following two schemes.

Poisson Sampling Denoted by $S^{\gamma}_{\textbf{Poisson}}$. $S^{\gamma}_{\textbf{Poisson}}$ takes a dataset of arbitrary size and

[3] $\mathcal{M}'$ can be adaptively chosen in that it could depend on the output of $\mathcal{M}$, which requires $\sup_{o \in \mathrm{Range}(\mathcal{M})} H_\alpha(\mathcal{M}'(D, o) \| \mathcal{M}'(D', o)) \leq H_\alpha(P' \| Q')$ for any value of o.

return a dataset by including *each* data point with probability $0 \leq \gamma \leq 1$ i.i.d. at random.

Subset Sampling Denoted by $S^{\gamma}_{\mathbf{Subset}}$. $S^{\gamma}_{\mathbf{Subset}}$ takes a dataset with size n or $n-1$ and return a subset of size $m < n$ uniformly at random. We define $\gamma := m/n$ as a short-hand. [4]

Somewhat unconventionally, the following theorem not only considers add/remove neighboring relation but also treat them *separately*, which turns out to be crucial in retaining a tight dominating pair with a closed-form expression. Our choice of choosing $\alpha \geq 0$ in Definition 3.3.1 ensures that for any mechanism (P, Q) dominates for add neighbors *iff* (Q, P) dominates for removal neighbors.

Theorem 3.3.5. *Let $\mathcal{M}$ be a randomized algorithm.*

(1) *If (P, Q) dominates $\mathcal{M}$ for add neighbors then $(P, (1-\gamma)P + \gamma Q)$ dominates $\mathcal{M} \circ S_{\mathbf{Poisson}}$ for add neighbors and $((1-\gamma)Q+\gamma P, Q)$ dominates $\mathcal{M} \circ S_{\boldsymbol{Poisson}}$ for removal neighbors.*

(2) *If (P, Q) dominates $\mathcal{M}$ for replacing neighbors, then $(P, (1-\gamma)P + \gamma Q)$ dominates $\mathcal{M} \circ S_{\mathbf{Subset}}$ for add neighbors and $((1-\gamma)P + \gamma Q, P)$ dominates $\mathcal{M} \circ S_{\mathbf{Subset}}$ for removal neighbors.*

We can obtain the results for the standard "add/remove" for a k-fold composition of subsampled mechanism by a pointwise maximum of the two:

$$\max\{H_{e^{\epsilon}}(P_1^k || Q_1^k), H_{e^{\epsilon}}(P_2^k || Q_2^k))\}$$

where (P_1, Q_1) is the "remove only" version of dominating pair and (P_2, Q_2) is the "add only" version of dominating pair.

[4] Note that here n, m are public and $\gamma := m/n$ even if $(n-1)$ is the sample size.

Existing literature that uses PLD for Poisson-sampled mechanisms while taking $(\gamma P + (1-\gamma)Q, Q)$ as an input are essentially providing privacy guarantees only for the "remove only" neighboring relationship. To the best of our knowledge, this is the first time a dominating pair of distributions under privacy-amplification by sampling is proven generically with an arbitrary base-mechanism $\mathcal{M}$ under the privacy-profile. The result, together with Theorem 3.3.4, allows PLD formalism to be applied to a broader family of mechanisms as well as their subsampled versions under adaptive composition.

Remark 3.3.6 (Comparing to subsampling results in RDP). Mironov et al. [2019] showed that $((1-\gamma)Q + \gamma P, Q)$ is also a dominating pair under the Renyi DP for $\mathcal{M} \circ S^{\gamma}_{\mathbf{Poisson}}$ if $\mathcal{M}$ is Gaussian mechanism. Zhu and Wang [2019] showed that there exits $\mathcal{M}$ under which $D_\alpha((1-\gamma)Q + \gamma P \| Q) < D_\alpha(Q \| (1-\gamma)Q + \gamma P)$, which suggests $((1-\gamma)Q + \gamma P, Q)$ is not always a dominating pair under RDP. Moreover, RDP of $\mathcal{M} \circ S^{\gamma}_{\mathbf{Subset}}$ are substantially trickier and it remains an open problem whether $((1-\gamma)Q + \gamma P, Q)$ is a dominating pair under RDP [Wang et al., 2019b].

3.3.2 Characteristic function representation

Having strengthened the foundation of the PLD formalism with "dominating distribution pairs" and two of its basic primitives, we can now put away RDP and its lossy (ϵ, δ)-DP conversion, then conduct mechanism-specific accounting under (ϵ, δ)-DP directly. Existing computational tools however, either require asymptotic approximation [Dong et al., 2021, Sommer et al., 2019], repeated convolution [Dong et al., 2021] or an *a priori* discretization of the output space [Koskela et al., 2020b]. This prompts us to ask:

"Can we compose mechanisms (with known dominating pairs) naturally just like in RDP? "

To achieve this goal, we propose using the *characteristic function* of the privacy loss RV.

Definition 3.3.7 (characteristic function of the privacy loss RV)**.** Let (P, Q) be a dominating pair of $\mathcal{M}$, and p, q be the probability density (or mass) function of P, Q. The two characteristic functions that describes the PLD are

$$\phi_{\mathcal{M}}(\alpha) : = \mathbb{E}_P[e^{i\alpha \log(p/q)}],\ \phi'_{\mathcal{M}}(\alpha) := \mathbb{E}_Q[e^{i\alpha \log(q/p)}],$$

where i denotes the imaginary unit satisfying $i^2 = -1$ and $\alpha \in \mathbb{R}$.

PLDs are probability measures on the real line, and these ϕ-functions are Fourier transforms of these measures. We provide ϕ-functions for basic mechanisms (see Table 3.2) and the discrete mechanisms with closed-form expression.

Advantages over MGF Comparing to the moment generating function used by the RDP, the characteristic function differs only in that we are taking the expectation of the *complex* exponential. At the price of bringing in complex arithmetics, it is now a complex-valued function supported on $\alpha \in \mathbb{R}$ rather than the real-valued Renyi Divergence with order $\alpha > 1$ as was defined in RDP. Unlike MGF, the characteristic function always exists and it characterizes the distribution of the privacy loss R.V., therefore it is always a *lossless* representation. MGF is also characteristic when it exists, but the conversion of MGF to the distribution function is numerically problematic [Epstein and Schotland, 2008].

Moreover, the adaptive composition over multiple heterogeneous mechanisms remains as straightforward as that of the RDP.

Proposition 3.3.8. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two randomized algorithms. We have the ϕ-function of the composition $(\mathcal{M}_1, \mathcal{M}_2)$ with order $\alpha \in \mathbb{R}$ satisfies: $\phi_{(\mathcal{M}_1,\mathcal{M}_2)}(\alpha) = \phi_{\mathcal{M}_1}(\alpha) \cdot \phi_{\mathcal{M}_2}(\alpha)$*

Lossless conversion rules. The ϕ-function can be losslessly converted back and

Mechanism	Dominating Pair	ϕ function
Randomized Response	$P : \Pr_P[0] = p; Q : \Pr_Q[1] = p$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = pe^{\alpha i \log(\frac{p}{1-p})} + (1-p)e^{\alpha i \log(\frac{1-p}{p})}$
Laplace Mechanism	$P : p(x) = \frac{1}{2\lambda}e^{-\lvert x\rvert/\lambda}; Q : q(x) = \frac{1}{2\lambda}e^{-\lvert x-1\rvert/\lambda}$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = \frac{1}{2}\left(e^{\frac{\alpha i}{\lambda}} + e^{\frac{-\alpha i-1}{\lambda}} + \frac{1}{2\alpha i+1}(e^{\frac{\alpha i}{\lambda}} - e^{\frac{-\alpha i-1}{\lambda}})\right)$
Gaussian Mechanism	$P : \mathcal{N}(1, \sigma^2); Q : \mathcal{N}(0, \sigma^2)$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = e^{\frac{-1}{2\sigma^2}(\alpha^2 - i\alpha)}$

Table 3.2: ϕ functions and dominating pairs for basic mechanisms.

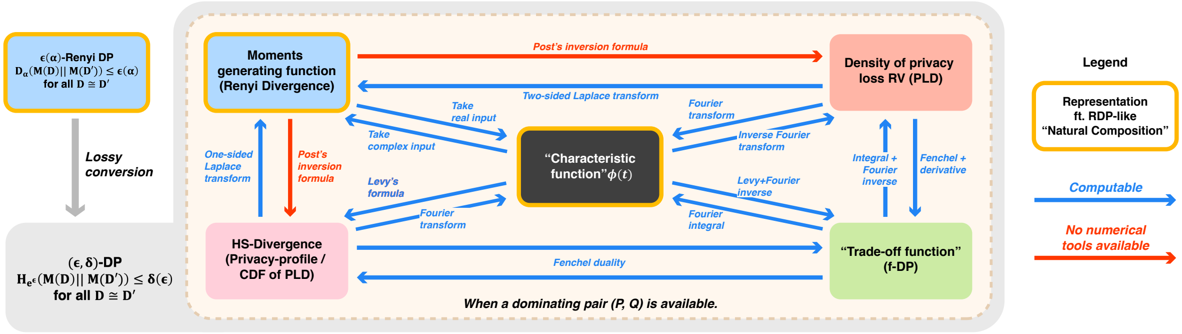

ϵ(α)-Renyi DP
Dα(M(D)|| M(D′)) ≤ ϵ(α)
for all D ≃ D′
Lossy conversion
(ϵ, δ)-DP
Hₑϵ(M(D)|| M(D′)) ≤ δ(ϵ)
for all D ≃ D′
Moments generating function (Renyi Divergence)
Post's inversion formula
Two-sided Laplace transform
Density of privacy loss RV (PLD)
Take real input
Take complex input
Fourier transform
Inverse Fourier transform
One-sided Laplace transform
Post's inversion formula
"Characteristic function" ϕ(t)
Integral + Fourier inverse
Fenchel + derivative
Levy's formula
Fourier transform
Levy+Fourier inverse
Fourier integral
HS-Divergence (Privacy-profile / CDF of PLD)
Fenchel duality
"Trade-off function" (f-DP)
When a dominating pair (P, Q) is available.
Legend
Representation ft. RDP-like "Natural Composition"
Computable
No numerical tools available

Figure 3.2: Summary of the various functional descriptions and their conversion rules.

forth with other representation such as the privacy-profile, tradeoff function, moment-generating function as well as the distribution function of the privacy loss RV. The conversion rule with prominent interest is the conversion to (ϵ, δ)-DP. Specifically, for finding δ as a function of ϵ (i.e., privacy profile), we invoke the fourth equivalent definition of (ϵ, δ)-DP in Lemma 3.1.4, which depends on the cumulative distribution function (CDF) of the privacy loss random variables $L_{P,Q}$ and $L_{Q,P}$. We can evaluate these CDFs through an integration of ϕ-functions via Levy's formula below.

Theorem 3.3.9 (Evaluate CDFs of privacy loss random variables)**.** *Let $\phi(\alpha)$ be characteristic function of privacy loss random variable $L_{P,Q}$. The CDF of $L_{P,Q}$ at point b satisfies*

$$F_{L_{P,Q}}(b) = \frac{1}{2} + \lim_{T\to\infty} \frac{1}{2\pi} \int_{-T}^{T} \frac{ie^{-i\alpha b}}{\alpha} \phi_{\mathcal{M}}(\alpha) d\alpha$$

The lossless conversions to other quantities are summarized in Figure 3.2. Moreover, most of the conversion formula correspond to well-known transforms such as the Fourier transform, Laplace transform and its double-sided variant. Except for those involve RDP and hence Laplace transform, numerical algorithms for implementing these transforms are often available.

3.4 Analytical Fourier Accountant and numerical algorithms

We now propose our analytical Fourier Accoutant (AFA) in Algorithm 1, which is a combination of the *lossless conversion rules* and the *analytical composition rule* (Proposition 3.3.8). Given a sequence of mechanisms (can be varied) applied to the same dataset, the data structure tracks the log characteristic function of each mechanism in a symbolic form. When there is a $\delta(\epsilon)$ query, the accountant first constructs two analytical CDFs (with respect to the privacy loss RV $L_{P,Q}$ and $L_{Q,P}$) using Theorem 3.3.9. Then the conversion to (ϵ, δ)-DP is obtained using Lemma 3.1.4. For computing ϵ given δ, we use bisection to solve $\delta_{\mathcal{M}}(\epsilon) = \delta$.

Algorithm 1 Analytical Fourier Accountant

```
Input Mechanisms $\mathcal{M}_1, ..., \mathcal{M}_K$ and $\delta$ .
for $i = 1, ..., K$ do
    Maintain the symbolic accountant
    $\log \phi_{(\mathcal{M})}(\alpha) \leftarrow \log \phi_{(\mathcal{M})}(\alpha) + \log \phi_{(\mathcal{M}_i)}(\alpha)$
    $\log \phi'_{(\mathcal{M})}(\alpha) \leftarrow \log \phi'_{(\mathcal{M})}(\alpha) + \log \phi'_{(\mathcal{M}_i)}(\alpha)$
    if query $(\epsilon, \delta)$-DP then
        Compute the CDF $F_{L_{P,Q}}(\cdot)$ and $F_{L_{Q,P}}(\cdot)$ by integrating $\log \phi_{(\mathcal{M})}(\alpha)$ and $\log \phi'_{(\mathcal{M})}(\alpha)$ using Theorem 3.3.9.
        Return $\delta$ by Lemma 3.1.4.
    end if
end for
```

AFA vs FFT. Comparing to the FFT-accountant approach [Koskela et al., 2020a,b, Koskela and Honkela, 2021], our approach decouples representation and numerical computation. We do not make any approximation when tracking the mechanisms, and use numerical computation only when converting to (ϵ, δ)-DP. This avoids the need for setting appropriate discretization parameters of FFT ahead of time before knowing which sequence $\mathcal{M}_1, ..., \mathcal{M}_K$ we will receive.

Gaussian quadrature For fast and numerically stable evaluation of the CDF, we propose to use Gaussian quadrature which adaptively selects the intervals between interpolation points, rather than the FFT approach which requires equally spaced discretization. When we apply this approach to efficiently evaluate integral in computing CDFs, where the numerical error is often negligible, i.e., $O(10^{-13})$ for CDFs in our experiments, even if we only sample a few hundreds points.

3.4.1 Experiments

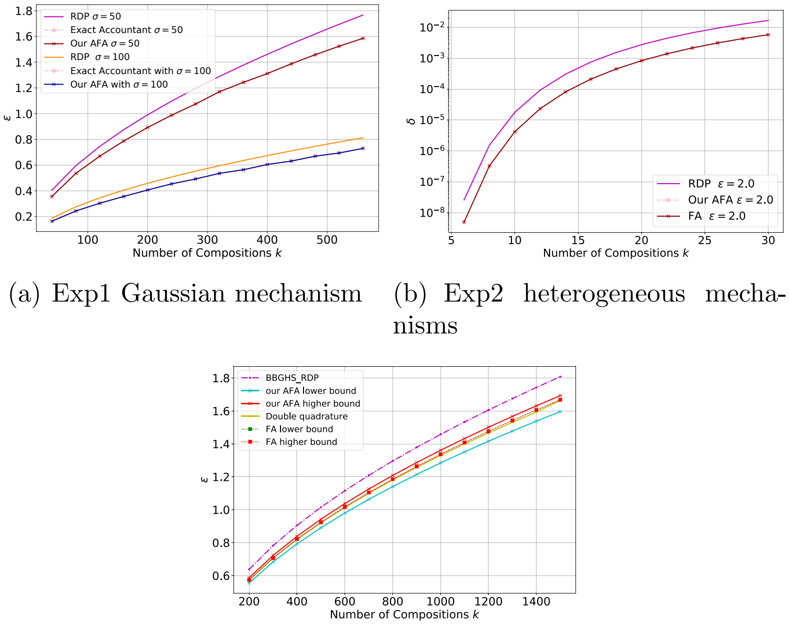
In this section, we conduct numerical experiments to illustrate the behaviors of our analytical Fourier Accountant. We will have three sets of experiments.

Exp. 1 (Gaussian mechanism) We compare the privacy cost over compositions between RDP accountant and AFA accountant on Gaussian mechanism.

Exp. 2 (Compositions of discrete and continuous mechanisms) We evaluate the Fourier accountant variants and RDP accountant on heterogeneous mechanisms.

Exp. 3 (Compositions over Poisson Subsample mechanisms) Comparison of our AFA with discretization-based ϕ-function to the Fourier accountant (FA) and the RDP accountant.

In Exp1, we compare our AFA method to the RDP-based accoutant[Mironov, 2017] and the exact accountant from the analytical Gaussian mechanism [Balle and Wang, 2018]. In Figure 3.3(a), we evaluate ϵ with a fixed $\delta = 10^{-4}$ and use $\sigma \in \{50, 100\}$.



(a) Exp1 Gaussian mechanism

(b) Exp2 heterogeneous mechanisms

(c) Exp3 Poisson Subsample

Figure 3.3: Pane 3.3(a) compares privacy cost over compositions in Exp 1. Pane 3.3(b) is for the heterogeneous composition in Exp 2. Pane 3.3(c) is for Poisson subsampled Gaussian mechanism in Exp 3.

Observation: In Figure 3.3(a), our ϕ function-based AFA exactly matches the result from the analytical Gaussian mechanism and strictly outperforms the RDP accountant in different privacy regimes.

In Exp2, motivated by [Koskela and Honkela, 2021], we consider an adaptive composition of the form $\mathcal{M}(X) = \Big(\mathcal{M}_1(X), \tilde{\mathcal{M}}_2(X), ..., \mathcal{M}_{k-1}(X), \tilde{\mathcal{M}}_k(X)\Big)$, where each $\mathcal{M}_i$ is a Gaussian mechanism with sensitivity 1, and each $\tilde{\mathcal{M}}_i$ is a randomized mechanism with probability p. We consider $\sigma = 5.0, p = 0.52, \epsilon = 2.0$ and compare $\delta(\epsilon)$ between the RDP accountant, Fourier Accountant [Koskela and Honkela, 2021] and our AFA.

Unlike the FA, our AFA allows an analytical composition over discrete and continuous mechanisms without sampling discretisation points over the privacy loss distribution, therefore achieves an exact privacy accountant. In Figure 3.3(b), we plot the $\delta(\epsilon)$ over k compositions given by FA and the moments accountant with RDP. We use $n = 10^5$

discretisations points and $L = 10$ for FA. Our numerical result matches FA as $n = 10^5$ is already a very accurate estimation as stated in [Koskela and Honkela, 2021].

There are cases when the closed-form ϕ-functions do not exist. In Exp 3, we consider this problem by analyzing the Poisson Subsample Gaussian mechanism using our discretization-based approach and "Double quadrature".

Figure 3.3(c) shows a comparsion of our AFA to the Fourier accountant method [Koskela et al., 2020b] and the moments accountant method [Zhu and Wang, 2019]. The sampling probability is $\gamma = 0.01$, the noise scale is $\sigma = 2.0$ and we evaluate ϵ with $\delta = 10^{-5}$. We use the tighter conversion rule from Balle et al. [2020] to convert the RDP back to (ϵ, δ)-DP. The numerical issues induced by Gaussian quadrature are at most $O(10^{-14})$. Our lower and upper bounds of $\delta(\epsilon)$ shown in Figure 3.3(c) already incorporate the error induced by discretization and ignoring the tail integral. We emphasize that the lower and upper bounds can match the bounds from FA by increasing sample points n. Moreover, "Double quadrature" is our proposed efficient approximation method. We only unevenly sample 700 points for each ϕ-function and the result of the "Double quadrature" lines between our lower and upper bounds and matches the result from FA. Lastly, all Fourier accountant-based approaches improve over the RDP-based accountant.

Runtime and space analysis of AFA We first compare the time complexity and memory when we have analytical expressions of ϕ-functions. In Exp 2, each mechanism admits an analytical ϕ-function and can be represented in $O(1)$ memory and evaluated in $O(1)$ time. Therefore, the memory cost is $O(\#\text{ unique mechanisms})$. We analyze the runtime by decomposing it into the "composition" and "conversion to $\delta(\epsilon)$" separately.

Let k denote the number of compositions. Regarding the runtime in the conversion to $\delta(\epsilon)$ query, we apply Gaussian quadrature to compute the CDF, which requires $O(\frac{1}{\delta_{err}^{1/\alpha}})$ runtime complexity for the αth order differentiable functions. The following composition runtime for Koskela and Honkela [2021] and Gopi et al. [2021] denote the runtime for discretization and convolution via FFT for a homogeneous composition of a mechanism for k rounds. We use n to denote the size of grid discretization in the FFT approximation.

Privacy accountant	Composition runtime	$\delta(\epsilon)$ conversion runtime	Memory	Choice of n
Our AFA	$O(1)$	$O(\frac{1}{\delta_{err}^{1/\alpha}})$	$O(1)$	Not applicable
Koskela and Honkela [2021]	$O(n\log n)$	$O(n\log n)$	$O(n)$	$n = O(k/\delta_{err})$
Gopi et al. [2021]	$O(n\log n)$	$O(n\log n)$	$O(n)$	$n = O(\sqrt{k}\log(1/\delta_{err})/\epsilon_{err})$

Table 3.3: The runtime/space complexity comparisons of different algoirthms

Of course, this is by no means a fair discussion because the FFT approach computes the entire (discretized) PLD of the composed mechanisms together while AFA computes just one point. In terms of the approximation error, our method is the only approach that adapts to the structures of the ϕ functions being integrated and achieves a faster convergence rate.

For the cases when the analytical expressions of ϕ-functions do not exist (see EXP3), we need to approximate the ϕ function too. Thus one single evaluation calls require $O(n)$, and our method is slower than Koskela and Honkela [2021], Gopi et al. [2021], because we do not use FFT. The space and time complexity of the adaptive discretization approach via double quadrature is unclear, though very fast in practice.

3.5 Conclusion

We studied the problem of privacy accounting with mechanism-specific analysis. We introduced the notion of *dominating pair distributions*, showed that each mechanism's privacy profile is *characterized* by a tight dominating pair, and derived a number of useful algebra of dominating pairs including *adaptive composition* and *amplification by sampling*. These results strengthen the foundation of the PLD formalism and make it more widely applicable. Algorithmically, we proposed an analytical Fourier accountant that represents the characteristic functions of a dominating pair *symbolically*, which features RDP-like natural composition and allows us to leverage off-the-shelf numerical tools. Our experiments demonstrate the merits of AFA and suggest that it can flexibly and efficiently fit into every DP application.

This work also leaves several open questions. Among those

- As Lemma 3.3.3 demonstrates, the construction of the domaining pair is severely constrained when trade-off functions are not clear. For example, characterizing high-dimension discrete Gaussian mechanism remains a tricky open problem.

- Moreover, there are cases where our approach requires much more quadrature points: We apply Gaussian quadrature to compute the CDF of the privacy loss RV through integration over ϕ-functions. If the composed ϕ functions have large values at the tail of integral (e.g., near ∞), we need to sample more quadrature points. We hope to solve this issue using numerical tools in the next step.

3.5.1 Omitted proofs in the main body

Characterization of privacy profiles

Proof: [Proof of Proposition 3.3.3] Let

$$\begin{aligned} \mathcal{H} &:= \{h : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0} \mid \exists P, Q \text{ s.t. } h(\alpha) = H_\alpha(P\|Q)\}, \\ \mathcal{F} &:= \{f : [0,1] \to [0,1] \mid \exists P, Q \text{ s.t. } f = T[P, Q]\}. \end{aligned}$$

By **??**, $H_\alpha(P\|Q)$ can be related to $f = T[Q, P]$ as follows:

$$H_{e^\varepsilon}(P\|Q) = 1 + f^*(-e^\varepsilon)$$

where ε ranges over the whole real line. By a simple change of variable, we see that $h \in \mathcal{H}$ iff there exists $f \in \mathcal{F}$ such that $h(\alpha) = 1 + f^*(-\alpha)$, or equivalently,

$$\mathcal{H} = \{h : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0} \mid \exists f \in \mathcal{F}, h(\alpha) = 1 + f^*(-\alpha)\}.$$

By Proposition 2.2 of Dong et al. [2021], we know

$$\mathcal{F} = \{f : [0,1] \to [0,1] \mid f \text{ is convex, decreasing, continuous and } f(x) \leqslant 1 - x\}.$$

Let $\mathcal{G} := \{g : (-\infty, 0] \to \mathbb{R} \mid g(0) = 0, g \text{ is convex, increasing, continuous and } g(x) \geqslant \max\{x, -1\}\}$.

<u>Claim:</u> Convex conjugacy is a bijection between $\mathcal{F}$ and $\mathcal{G}$. *Proof:* [Proof of the claim] Since both $\mathcal{F}$ and $\mathcal{G}$ consist of convex functions, double convex conjugacy brings back the function, it suffices to show that $f \in \mathcal{F} \implies f^* \in \mathcal{G}$ and $g \in \mathcal{G} \implies g^* \in \mathcal{F}$. Now suppose $f \in \mathcal{F}$. f is extended to be $+\infty$ in $(-\infty, 0)$ and 0 in $(1, +\infty)$. Thus f is a convex function on $\mathbb{R}$. By definition f^* is convex, and we can calculate

$$f^*(y) = \sup_{x \in \mathbb{R}} yx - f(x) = \sup_{x \geqslant 0} yx - f(x) = \begin{cases} +\infty, \text{ if } y > 0 \\ 0, \text{ if } y = 0 \end{cases}$$

With $y_1 < y_2$, we have $y_1 x - f(x) \leqslant y_2 x - f(x)$. Taking supremum over $x \geqslant 0$, we have $f^*(y_1) \leqslant f^*(y_2)$. This shows f^* is monotone and finite on $(-\infty, 0]$. Let

$$I(x) = \begin{cases} +\infty, \text{ if } x < 0 \\ \max\{1 - x, 0\}, \text{ if } x \geqslant 0 \end{cases}$$

It is straightforward to compute that

$$I^*(y) = \begin{cases} \max\{y, -1\}, & \text{if } y \leqslant 0 \\ +\infty, & \text{if } y > 0 \end{cases}$$

Since $f \leqslant I$, we conclude that $f^*(x) \geqslant I^*(x) = \max\{x, -1\}$.

Now suppose $g \in \mathcal{G}$. Similarly, g is extended to be $+\infty$ in $(0, +\infty)$. $g^*(y) = \sup_{x \leqslant 0} yx - g(x)$ and $g^*(y) = +\infty$ if $y < 0$. By a similar argument, g^* is increasing. Since $g \geqslant I^*$, we have $g^* \leqslant I^{**} = I$. That is, $g^*(x) \leqslant 1 - x$. Let J be zero on $(-\infty, 0]$ and infinity otherwise. We have J^* is zero on $[0, +\infty)$ and infinity otherwise. We know that $g(0) = 0$ and g is increasing so $g \leqslant J$. Hence $g^* \geqslant J^*$, i.e. $g^*(y) \geqslant 0$ if $y \geqslant 0$. This justifies that $g^*(y) \in [0, 1]$ if $y \in [0, 1]$ and $g^*(y) = 0$ if $y \geqslant 1$. ■

Now with the help of this claim, $\mathcal{H}$ and $\mathcal{G}$ are simply related: $h \in \mathcal{H}$ iff $\alpha \mapsto h(-\alpha) - 1$ is in $\mathcal{G}$. Therefore we can get the description of $\mathcal{H}$. The proof of the first statement is complete.

Explicit construction. Next we derive the specific choice of P, Q as stated works using the result from Dong et al. [2021].

Continuing with the notations in the proof above, when H satisfies the conditions, i.e. $H \in \mathcal{H}$, we know there is a $f \in \mathcal{F}$ such that $H(\alpha) = 1 + f^*(-\alpha)$. Let $g(\alpha) = H(-\alpha) - 1$ and we will have $g = f^*$ and hence $f = g^*$ as f is convex. Therefore,

$$f(x) = g^*(x) = \sup_y yx - H(-y) + 1 = \sup_z -zx - H(z) + 1 = 1 + H^*(-x).$$

From Dong et al. [2021, Proposition 2.2], we know that $f = T[Q, P]$ where $Q = U[0, 1]$

is the uniform distribution over $[0,1]$ and P has CDF

$$F_P(x) = \begin{cases} 0, \text{ if } x < 0, \\ f(1-x), \text{ if } x \in [0,1), \\ 1, \text{ if } x \geqslant 1. \end{cases}$$

Plugging in $f(x) = 1 + H^*(-x)$, we have the CDF of P being

$$F_P(x) = \begin{cases} 0, \text{ if } x < 0, \\ 1 + H^*(1-x), \text{ if } x \in [0,1), \\ 1, \text{ if } x \geqslant 1. \end{cases}$$

Note that when the infimum of H is positive, $H^*(1-x) < 0$ and P has an atom at 1. This completes the proof. ■

Another interesting consequence of Lemma 3.3.3 is one can often get a stronger bound on the hockey-stick divergence or privacy profile *for free*. Recall that for a function g, its convex hull $\text{conv}(g)$ (a.k.a., the lower convex envelope) is defined as the greatest convex lower bound of g and satisfies $\text{conv}(g) = g^{**}$ where the double star means taking Fenchel conjugate twice.

For a function $h : \mathbb{R}_+ \to \mathbb{R}$, let $g(x) = \inf_{y \in [0,x]} h(y)$ and $\text{HS}(h) = (\min\{1, g\})^{**}$. It turns out that $\text{HS}(h)$ is the greatest lower bound of h that lies in $\mathcal{H}$, and we have

Corollary 3.5.1 (Dominating pairs from any privacy profile upper bounds)**.** *If the privacy profile of a mechanism $\mathcal{M}$ is bounded by $h : \mathbb{R}_+ \to \mathbb{R}$, i.e. $\delta_{\mathcal{M}}(\alpha) \leqslant h(\alpha), \forall \alpha \geqslant 0$, then $\delta_{\mathcal{M}}$ is also bounded by* $\text{HS}(h)$.

Note that $\text{HS}(h)$ can be significantly smaller than the original bound h, and it admits a dominating pair by Proposition 3.3.3, even if h does not.

Proof: We know that $\delta_{\mathcal{M}} \in \mathcal{H}$. It suffices to show that

$$f \in \mathcal{H}, f \leqslant h \implies f \leqslant \mathrm{HS}(h).$$

Recall that we let $g(x) = \inf_{y\in[0,x]} h(y)$ and $\mathrm{HS}(h) = (\min\{1, g\})^{**}$. Since $f \in \mathcal{H}$ is decreasing, $f(x) = \inf_{y\in[0,x]} f(y) \leqslant \inf_{y\in[0,x]} h(y) = g(x)$. Furthermore, $f(x) \leqslant f(0) = 1$, so $f \leqslant \min\{1, g\}$. Since f is convex, it also holds that $f \leqslant \min\{1, g\}^{**} = \mathrm{HS}(h)$. ■

Composition theorem of dominating pairs

Theorem 3.5.2 (Restatement of Theorem 3.3.4 Adaptive composition of dominating pairs)**.** *Let P, Q be a dominating pair distributions for $\mathcal{M}$ and P', Q' be a dominating pair distributions for $\mathcal{M}'$*[5]*, then $(P \times P', Q \times Q')$ is a dominating pair distributions for the composed mechanism $(\mathcal{M}, \mathcal{M}')$.*

Proof:

$$H_\alpha(P\|P') = \int_\Omega [p(\omega) - \alpha p'(\omega)]_+ \, \mathrm{d}\omega.$$

Integration with respect to a dominating measure of both P and Q and p, q are the densities (Radon-Nikodym derivatives) for the probability measures P, Q respectively.

Our goal is to show $H_\alpha\big(M(D), M(D')\big) \leqslant H_\alpha\big(P \times R, Q \times S\big)$. We break it into the following two parts.

$$H_\alpha\big(M(D), M(D')\big) \leqslant H_\alpha\big(M_1(D) \times R, M_1(D') \times S\big) \leqslant H_\alpha\big(P \times R, Q \times S\big).$$

[5] $\mathcal{M}'$ can be adaptively chosen in that it could depend on the output of $\mathcal{M}$, which requires $\sup_{o\in \mathrm{Range}(\mathcal{M})} H_{e^\epsilon}(\mathcal{M}'(D, o)\|\mathcal{M}'(D', o)) \leq H_{e^\epsilon}(P'\|Q')$ for any value of o.

Starting from the first part, we have

$$\begin{aligned}
H_\alpha\big(M(D), M(D')\big) &= \iint_{X\times Y} [p_1(x)p_2(x,y) - \alpha p_1'(x)p_2'(x,y)]_+ \,\mathrm{d}x\,\mathrm{d}y \\
&= \int_X p_1(x)\cdot\left(\int_Y \left[p_2(x,y) - \alpha\cdot\frac{p_1'(x)}{p_1(x)}\cdot p_2'(x,y)\right]_+ \mathrm{d}y\right)\mathrm{d}x \\
&= \int_X p_1(x)\cdot\left(H_{\alpha\cdot\frac{p_1'(x)}{p_1(x)}}\big(M_2(D,x)\|M_2(D',x)\big)\right)\mathrm{d}x \\
&\leqslant \int_X p_1(x)\cdot\left(H_{\alpha\cdot\frac{p_1'(x)}{p_1(x)}}\big(R\|S\big)\right)\mathrm{d}x \\
&= \int_X p_1(x)\cdot\left(\int_{\Omega_2} \left[r(\omega_2) - \alpha\cdot\frac{p_1'(x)}{p_1(x)}\cdot s(\omega_2)\right]_+ \mathrm{d}\omega_2\right)\mathrm{d}x \\
&= \iint_{X\times\Omega_2} [p_1(x)r(\omega_2) - \alpha p_1'(x)s(\omega_2)]_+ \,\mathrm{d}x\,\mathrm{d}\omega_2 \\
&= H_\alpha\big(M_1(D)\times R, M_1(D')\times S\big).
\end{aligned}$$

Continuing this argument, we have

$$\begin{aligned}
H_\alpha\big(M_1(D)\times R, M_1(D')\times S\big) &= \iint_{X\times\Omega_2} [p_1(x)r(\omega_2) - \alpha p_1'(x)s(\omega_2)]_+ \,\mathrm{d}x\,\mathrm{d}\omega_2 \\
&= \int_{\Omega_2} r(\omega_2)\cdot\left(\int_X \left[p_1(x) - \alpha\cdot\frac{s(\omega_2)}{r(\omega_2)}\cdot p_1'(x)\right]_+ \mathrm{d}x\right)\mathrm{d}\omega_2 \\
&= \int_{\Omega_2} r(\omega_2)\cdot\left(H_{\alpha\cdot\frac{s(\omega_2)}{r(\omega_2)}}\big(M_1(D)\|M_1(D')\big)\right)\mathrm{d}\omega_2 \\
&\leqslant \int_{\Omega_2} r(\omega_2)\cdot\left(H_{\alpha\cdot\frac{s(\omega_2)}{r(\omega_2)}}\big(P\|Q\big)\right)\mathrm{d}\omega_2 \\
&= \int_{\Omega_2} r(\omega_2)\cdot\left(\int_X \left[p(\omega_1) - \alpha\cdot\frac{s(\omega_2)}{r(\omega_2)}\cdot q(\omega_1)\right]_+ \mathrm{d}\omega_1\right)\mathrm{d}\omega_2 \\
&= \iint_{\Omega_1\times\Omega_2} [p(\omega_1)r(\omega_2) - \alpha q(\omega_1)s(\omega_2)]_+ \,\mathrm{d}\omega_1\,\mathrm{d}\omega_2 \\
&= H_\alpha\big(P\times R, Q\times S\big).
\end{aligned}$$

The proof is complete. ■

Privacy-amplification for dominating pairs

Recall we stated the following theorem in the main body:

Remark 3.5.3 (Exact optimality of the bounds). If (P, Q) is a *tightly dominating pair* for $\mathcal{M}$, for both "Removal"-neighboring relation or "Add"-neighboring relation, then under some mild regularity conditions on $\mathcal{M}$ and the space of the input datasets, Theorem 3.3.5 can be strengthened to show that that $((1-\gamma)Q + \gamma P, Q)$ and $(P, (1-\gamma)P + \gamma Q)$ are *tight dominating pairs* for the "Removal"-neighboring relation and "Add"-neighboring relation respectively — i.e., the dominating pair is realized by some concrete datasets. For example, consider $\mathcal{M}$ to be Gaussian mechanism or Laplace mechanism that releases the total number of 1s in a dataset. Then two neighboring datasets $D = [0, ..., 0, 0, 1] \in \mathbb{R}^{n+1}$, $D' = [0, 0, ..., 0] \in \mathbb{R}^n$ for "removal" and $D = [0, ..., 0] \in \mathbb{R}^n$, $D' = [0, 0, ..., 0, 1] \in \mathbb{R}^{n+1}$ for "addition" attains the upper bound for all $\alpha > 0$ in each category.

Remark 3.5.4 (Renyi DP and Optimal Moments Accountant for subsampled mechanisms). Renyi-DP and moments accountant are closely related concepts that are often considered identical. However, our results suggest that there is a distinction. The above pair of P, Q we constructed are not necessarily attaining the Renyi-DP bounds (see a concrete example from Zhu and Wang [2019], but as moments accountant focuses only on computing (ϵ, δ)-DP, it suffices use the Renyi-divergence functions $R_\alpha(P\|Q)$. Specifically, this closes the constant gap between the moments accountant for subsampled mechanisms and Poisson sampled mechanisms.

Part II

Modern privacy accountings with new private deep learning methods

Chapter 4

Private-kNN: practical differential privacy for computer vision

4.1 Introduction

The key idea of differentially private machine learning is to appropriately *randomize* the training process (e.g. adding noise), so the fitted model parameters can be thought of as a sanitized "release" with individual information removed. Most existing approaches do not apply for deep learning [Chaudhuri et al., 2011, Dimitrakakis et al., 2014, Wang et al., 2015, Park et al., 2016]. A notable exception — NoisySGD [Song et al., 2013, Bassily et al., 2014, Abadi et al., 2016] — requires privately releasing the gradients for many iterations by adding noise proportional to $\sqrt{d}$ to every coordinate of the gradient in a model with d-parameters, hence does not scale to the large models with millions of parameters that are commonly used in computer vision.

A recent *model agnostic* approach, termed "Private Aggregation of Teacher Ensembles" (PATE), introduces a model aggregation strategy and gains privacy by injecting randomness into the aggregation. It assumes a teacher-student knowledge transfer frame-

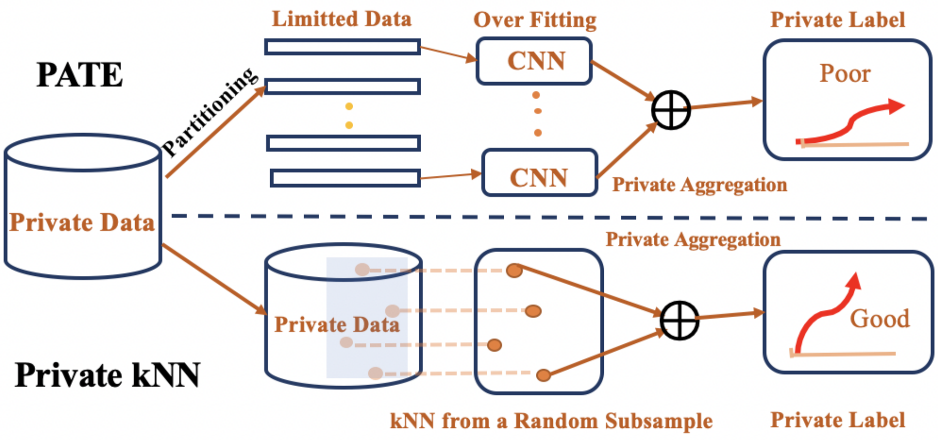


Figure 4.1: A comparison of PATE's framework and ours.

work by leveraging an isolated private data and unrestricted public unlabeled data. The most critical parameter to choose in PATE is the number of teachers k. It largely determines the margin between top vote and the second top vote, i.e., k is often as large as 250 for a meaningful privacy guarantee while ensuring the pseudo labels are sufficiently accurate. As known, each teacher deep learning model requires sufficient amount of data to generalize well due to the neural network's data-starving property. While in practice, it is common that data, especially private data is very limited, which cannot support partitioning into many disjoint piles. In CIFAR-10, if set $k = 250$, each teacher is assigned with only 200 images and can achieve accuracy below 50%.

To address the problem, we propose a more data-efficient differential private algorithm based on releasing the pseudo-labels using the majority voting of the k-nearest neighbors (kNN). This approach avoids data-splitting because adding or removing an individual to the data can change at most one of anybody's k-nearest neighbor. This enables us to choose larger k without worrying about not having enough data to train teachers — kNN involves no training at all. Moreover, this allows is to leverage the recent advances in "privacy amplification by sampling" to label orders-of-magnitude more public data with only a fraction of the cost in privacy loss than PATE.

Careful readers may legitmately ask: how does this allow us to take advantage of the modern deep learning? Despite the strong guarantee of kNN that says it asymptotically achieves the Bayes rate [Cover and Hart, 1967], it is not known as a state-of-the-art classifier in finite-sample computer vision problems. Our novel solution to this problem to make learning *interactive*. Specifically, we outsource the representation-learning task to the public domain where the student, trained with your favorite deep learning model, will share the learned feature map with the teacher, so the quality of kNN's pseudo-labels will improve, which in turn, helps the student to learn a better representation as we iterate. It is worth noting that the use of "privacy amplification by sampling" is central in our design, as it buys us the necessary adaptivity through allowing the ability to release many pseudo-labels.

Our main contributions are summarized below:

1. We propose Private k-Nearest Neighbor (kNN) for differentially private (DP) deep learning under the "knowledge transfer" framework. It represents the first practical solution that addresses this important problem scales to large models while preserving theoretically meaningful DP guarantee ($\epsilon < 1$).

2. We present a new Renyi-differential privacy analysis to a "noisy screening" mechanism proposed in [Papernot et al., 2018]. This allows us to use it with the moments accountant for a tighter privacy accounting. Collectively, "subsampling" and "noisy screening" allows us to answer 10 times more queries with even less privacy budget compared to state-of-the-art PATE models. The data-dependent version of this "Noisy screening" mechanism can be thought of as a post-hoc Gaussian-noise version of the celebrated Sparse Vector Technique in differential privacy, and is of independent interest.

3. We examine our approach with extensive vision tasks such as MNIST, SVHN, CIFAR-10 and also evaluate on two realistic identity relevant tasks, namely, face attribute classification on Celeb-A and human body attribute classification on Market1501. Private-kNN achieves consistently better performance across privacy cost and accuracy in all the benchmarks when compared to the state-of-the-art differential privacy learning methods.

4.2 Preliminary

Privacy Amplification by Subsampling. Subsampling is a widely used algorithmic tool in privacy, which deals with a composite mechanism that first randomly samples the data, and then applies a DP mechanism on the randomly selected subset. Intuitively, since the one person that differs between X and X' is often not selected in the subset, the overall privacy guarantee should be stronger. Loosely speaking, when we apply an (ϵ, δ)-DP mechanism to a random γ-proportion of the data, the whole procedure satisfies $(O(\gamma\epsilon), \gamma\delta)$-DP. The result of this style is also known as "subsampling lemma" or "secrecy of the samples" in the literature Balle et al. [2018]. This is practically relevant as it is the reason why we can afford to run Noisy-SGD Song et al. [2013] for many iterations without blowing up the privacy cost. Recently, such as "subsampling lemma" was proven for the RDP. The benefits of the subsampling can be combined with the tight advanced composition of RDP [Wang et al., 2019b, Zhu and Wang, 2019], which roughly says that under some restrictions on α:

$$\epsilon_{\mathcal{M}\circ\mathrm{Sample}_\gamma}(\alpha) \leq O(\gamma^2 \epsilon_{\mathcal{M}}(\alpha)).$$

In this work, we apply a Poisson subsampled "RDP-amplification " bound from Zhu and Wang [2019]. A more precise statement of this result is attached in the appendix. We emphasize that this is the main technical contribution leveraged in this work that simply cannot be done under the PATE approach.

Data-Dependent RDP and PATE The privacy analysis in PATE is straight-forward. It involves injecting Laplace noise [Papernot et al., 2017] or Gaussian noise [Papernot et al., 2018] to the teacher votes. For noise with standard deviation $O(k)$, a budget of ϵ, δ, roughly speaking, allows PATE to release $O(\frac{\epsilon^2 k^2}{\log(1/\delta)})$ pseudo-labels, which is insufficient for many cases.

A notion of data-dependent RDP is introduced to further take into account of the high margin that occurs when the teachers largely agree with each other, in which case the privacy cost is intuitively smaller.

Definition 4.2.1 (Data-dependent RDP [Papernot et al., 2017])**.** A mechanism $\mathcal{M}$ is (α, ϵ)-data-dependent RDP with order $\alpha \in (1, \infty)$ if for all X' that is a adjacent to X

$$\max\{D_\alpha(\mathcal{M}(X)\|\mathcal{M}(X')), D_\alpha(\mathcal{M}(X')\|\mathcal{M}(X))\} \leq \epsilon.$$

In other words, the data-dependent *RDP function* ϵ is a joint function of X and α. There are a few other tricks proposed in [Papernot et al., 2018] to reduce the total privacy loss. Notably, they designed a " noisy screen" step that first adds a larger Gaussian noise to max{votes}, and then release a more confident version of votes only for those questions that passes the screening. This allows PATE to save privacy loss via data-dependent RDP in the second step with smaller noise. In this paper, we use the same "noisy screening" but provide a tighter analysis of this procedure that saves a constant fraction of the privacy budget.

Finally, we note that the use of data-dependent RDP can be seen as controversial,

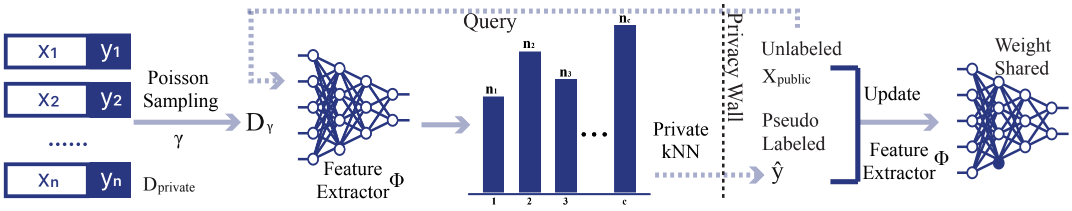


Figure 4.2: The overview of the proposed framework. Given the unlabeled public data X_{public}, we query through privacy wall for pseudo labels, where the private data and the queried public data are sent through feature extractor Φ and "Private-kNN" to assign pseudo labels. Combining the public data and the pseudo labels, the feature extractor Φ is further updated. This procedure can be iterated for rounds to achieve satisfied privacy-accuracy trade-off.

as the resulting privacy loss ϵ is now a sensitive quantity that depends on the data. [Papernot et al., 2018] provided a smooth-sensitivity based method [Nissim et al., 2007] to privately release $\epsilon_{\mathcal{M},X}(\alpha)$ for a sequence of α, but that incurs additional privacy losses that are not reported in their main result. One major contribution of the current paper is to demonstrate that practical differential privacy can be achieved when training a deep networks under the "knowledge transfer" setting even without using data-dependent RDP.

4.3 Our Approach

We are now ready to describe our method: Private-kNN.

Notations and symbols. In this section and thereafter, we stick to the following notations. $x \in \mathcal{R}^d$ denotes the feature of both private and public data. Let $D_{private}$ be the private dataset of size n: $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ and $y_i \in [1, c]$ is the label, where c is the number of classes in $D_{private}$. Let m be the size of the unlabeled public data. γ is the sampling ratio used to sample a random subset D_γ from $D_{private}$. We define ϕ be the feature extractor for private kNN. $f_j(x)$ is the prediction of j^{th} neighbor on the public feature x and the total number of neighbors is k. In the noisy screening, we use

σ_1 to denote the Gaussian noise scale, and T is the threshold for a screening check. σ_2 is the Gaussian noise scale for the noisy aggregation procedure. ϵ and δ are reserved for denoting privacy cost.

Setup. As defined in PATE, we have access to a private dataset and an unlabeled public dataset, and we seek to design an (ϵ, δ)-DP algorithm that outputs pseudo-labels for as much public data as possible. Then a student model is trained via semi-supervised learning using both pseudo labeled and unlabeled public data. Again, by the property of "closedness to postprocessing", the student model itself satisfies DP assumption.

Private-kNN. Our algorithm involves four simple steps.

1. Pick k-Nearest Neighbors with Poisson Sampling For each query x from the public domain, we use Poisson sampling[1] to get a random subset from the entire private dataset. Then we pick the k nearest neighbors from $\mathcal{D}_\gamma$ by measuring their Euclidean distance in feature space $\mathcal{R}^{d_\phi}$, where ϕ is a non-private feature extractor. The choice of Euclidean distance is general, whereas other distance metrics can also be applied. Our algorithm is designed into rounds of iterations. In the first iteration, ϕ is initialized with a Histogram of Oriented Gradient (HOG)Dalal and Triggs [2005] feature extractor, which is a popular descriptor used in the computer vision tasks. In the next iteration, we apply a deep neural network for the public student model (except for the last softmax layer) to update the feature extractor ϕ. In the experiment section, we show how this interactive scheme iteratively refines the feature embedding used by Private-kNN.

2. Noisy Screening. let $f_j(x)$ be the prediction of j^{th} neighbor on x, where $j \in [1, k]$. The label count of class $i \in [1, c]$ is

$$n_i(x) = |\{j : f_j(x) = i\}|$$

[1] Possion sampling includes each data point independently with probability γ. It can be efficiently implemented by first sample the size of the subset from a Binomial distribution then find a random subset.

Answering all queries from public without selection leads to running out privacy budget instantly. To be more selective, we only answer those queries which have an overwhelming consensus in voting, and this screening process is implemented privately with Gaussian noise parameter σ_1, for the query not passing the noisy screening check, we return $\perp$, and ignore this data in re-training a student model.

$$\text{If } \max_i\{n_i(x)\} + \mathcal{N}(0, \sigma_1^2) \leq T \text{ then return } \perp$$

T here is the threshold parameter for screening, we set $T \approx 0.6 \times k$ in the hope of there is consensus among neighbors upon this query. Since we pay for private screening for every query, a larger σ_1 would be helpful for privacy concerns. As we mentioned before, the same screening procedure is used in PATEPapernot et al. [2018] and despite a larger noise, this is still the most costly part of PATE. PATE treated this screening procedure as a simple post-processing of the Gaussian mechanism. We note that the output is actually drawn from a discrete distribution of either $\top$ (Pass) or $\perp$ (Fail). In the next section we derive the RDP for this procedure, which allows to benefit from moments accountant.

3. Noisy Aggregation For those query x which pass the check, we release its label

$$f(x) = \arg\max_j\{n_j + \mathcal{N}(0, \sigma_2^2)\}$$

with a fresh random subsample of the data. The noisy screening process filters out about 50% query, which enables the noisy aggregation process to have a smaller σ_2 for better-aggregated accuracy.

4.Training Student Model Our model only answers a selected number of queries from the public. Otherwise the final privacy cost becomes meaningless. Taking the answered queries as pseudo labeled data, together with the unlabeled data, a student

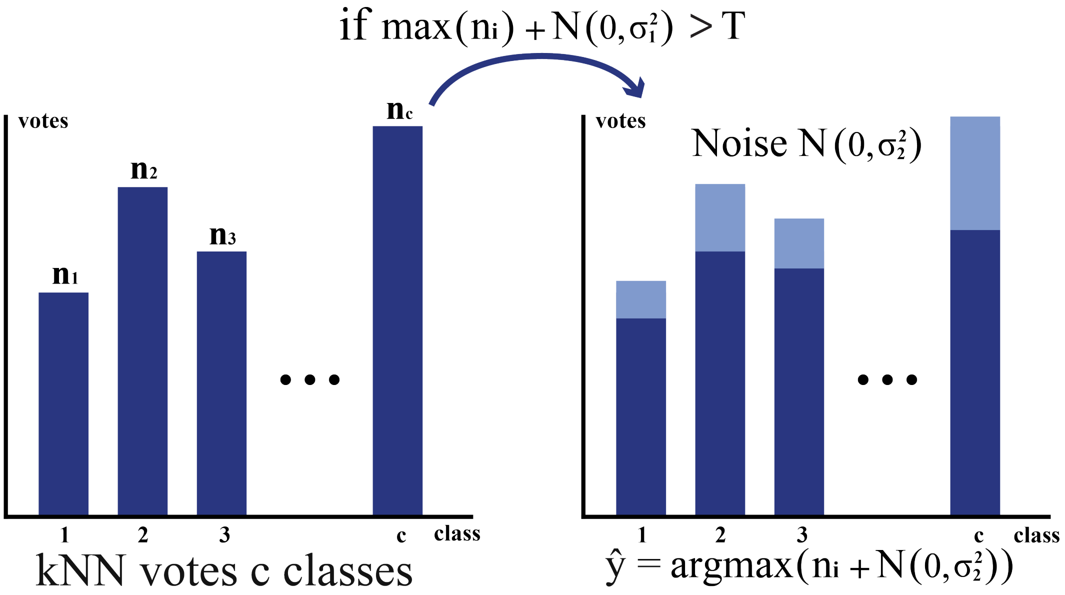


Figure 4.3: Illustration on the noisy screening and noisy aggregation procedure.

model is trained in the self-supervised manner. We consider two popular self-supervised methods: virtual adversarial training(VAT)Miyato et al. [2018] and unsupervised Data Augmentation(UDA)Xie et al. [2019]. VAT uses the virtual adversarial perturbation in the noisy process and UDA exploits advanced data augmentation instead of random augmentation. In our experiments, we find that UDA outperforms VAT in both SVHN and CIFAR-10 tasks. As shown in Figure 4.2, the student model is trained with the above mentioned self-supervised method. On the other hand, the student model is utilized to extract the updated feature in the private domain for private-kNN. This iterative feature distilling allows private-kNN to have similar capacity as ConvNet (replace the last softmax layer in ConvNet with kNN), and to further improve the accuracy of answering public queries. Besides, iterative training allows to exploit the benefits from unlabeled public data, which does not violate the DP assumption or incur any privacy cost, but is shown to enhance the utility of student model under the self-supervised training.

Privacy analysis. We prove the DP guarantee in the following. Let $\mathcal{M}$ denote the mechanism of Private-kNN. Our method can be viewed as a composition of $(\mathcal{M}_s) \circ \mathrm{Sample}_\gamma$ and $(\mathcal{M}_{\sigma_2}) \circ \mathrm{Sample}_\gamma$. Based on composition theorem, the privacy cost can be traced by individually calculating the RDP of the two mechanisms and then add them up. For the latter, we can readily apply the tight bound of the sub-sampled Gaussian mechanism from [Zhu and Wang, 2019]. Our main theoretical result is the following characterization of the noisy screening procedure via a tight RDP analysis.

Theorem 4.3.1 (RDP of "Noisy Screening")**.** *Let $\mathcal{M}_s$ be a randomized algorithm for noisy screening procedure with a predefined Gaussian noise scale σ_1 and the threshold T. Then $\mathcal{M}_s$ obeys RDP with*

$$\epsilon_{\mathcal{M}_s}(\alpha) = \max_{(p,q)\in\mathcal{S}} \frac{1}{\alpha - 1} \log\left(p^\alpha q^{1-\alpha} + (1-p)^\alpha (1-q)^{1-\alpha}\right).$$

where $\mathcal{S}$ contains the following "pairs":

$$\left(\mathbb{P}[\mathcal{N}(t,\sigma_1^2) \geq T], \mathbb{P}[\mathcal{N}(t+1,\sigma_1^2)] \geq T]\right),$$
$$\left(\mathbb{P}[\mathcal{N}(t,\sigma_1^2) \geq T], \mathbb{P}[\mathcal{N}(t-1,\sigma_1^2)] \geq T]\right)$$

for all integer $\lceil k/c \rceil \leq t \leq k$.

We remark that the above bound can be calculated efficiently for any pairs of k, T in $O(k)$ time and can be evaluated by calculating the Gaussian cumulative density function using the efficient implementation of the error function erfc. A more detailed proof is provided in the appendix. Moreover, it is more numerically stable to directly represent the log of p and q above. By the information-processing inequality of Rényi-divergence, this bound is strictly better than that from the Gaussian mechanism for every α.

Finally, we estimate the overall privacy bound for the end-to-end method.

Theorem 4.3.2 (Asymptotic scaling)**.** *The total privacy bound of Private-kNN to label* *all m public data points with noise σ_1, σ_2 is (ϵ, δ)-DP, with any δ, and*

$$\epsilon = O(\gamma\sqrt{\log(1/\delta)}(\frac{\sqrt{m}}{\sigma_1} + \frac{\sqrt{m_{selected}}}{\sigma_2})).$$

The proof is in the appendix. Notice that this is only used for illustrating the amplification effect γ that is not present in PATE. The actually numerical calculation of ϵ is tighter using analytical moments accountant [Wang et al., 2019b].

4.4 Experiments

In this section, we demonstrate our Private-kNN for its data efficiency with character recognition tasks such as MNIST LeCun et al. [1998] and SVHN Netzer et al. [2011]. We show that our model achieves the same accuracy with only 10% of the privacy cost used in state-of-the-art (SOTA) methods such as PATE Papernot et al. [2017]. We also leverage the general vision tasks where data splitting for PATE is the bottleneck. CIFAR-10 Krizhevsky [2009] as a general object recognition task is investigated across the DP methods. More specifically, we focus on two realistic setting vision problems, namely face attribute classification on CelebA Liu et al. [2015] and body attribute classification on Market1501 Zheng et al. [2015], which is the first to show that our method can facilitate to realistic multi-label classification tasks.

MNIST and SVHN Evaluation

MNIST and SVHN are two common datasets to measure the utility and privacy performance of differential private models Papernot et al. [2017, 2018]. We evaluate Private-kNN using the same setup of private dataset and the model architecture as in PATE Papernot et al. [2017, 2018]. On MNIST, the training set is reserved as the

Table 4.1: Utility and privacy of semi-supervised student model

Dataset	Methods	#Queries	ϵ	Acc.	NP Acc.
MNIST	LNMAX	1000	8.03	98.1%	99.2%
	GNMAX	286	1.97	98.5%	
	Ours	735	**0.47**	98.8%	
SVHN	LNMAX	1000	8.19	90.1%	92.8%
	GNMAX	3098	4.96	91.6%	
	Ours	2939	**0.49**	91.6%	
CIFAR-10	GNMAX			$\leq 50\%$	80.5%
	Noisy SGD		4	70%	
	Ours	3877	**2.92**	70.8%	

Table 4.2: Ablative results of iterative training on SVHN dataset.

Iteration	kNN Acc.	retrain CNN	#Queries	ϵ
1	82.5%	86.6%	1022/3000	0.49
2	94.41%	91.6%	1917/3000	

private dataset, half of the testing set acts as unlabeled student training data, and the remaining part is for real testing. For SVHN, the extended data, together with training data, are regarded as private data. Among the $26k$ testing set, $25k$ acts as publicly unlabeled student data for query and self-supervised training, where the remaining $1k$ is for testing. We defer the detailed information of model architectures in appendix and report their non-private baselines in Table 4.1.

As illustrated in the method, we conduct initial round kNN classification using a hand-crafted feature — histogram of oriented gradients(HOG). Then we apply self-supervised training (e.g.Miyato et al. [2018], Xie et al. [2019]) with the pseudo-labeled data from kNN for better feature representation learning.

MNIST: In our method, the privacy cost is accumulated over 1000 queries of 2 iterations. We set the number of neighbors $k = 300$, $\sigma_1 = 75$ for screening, threshold $T = 180$ and $\sigma_2 = 25$ for aggregation, and fix the sub-sampling ratio $\gamma = 0.15$. In the initial iteration, the accuracy of the privately aggregated kNN model based on HOG feature is 92.1%.

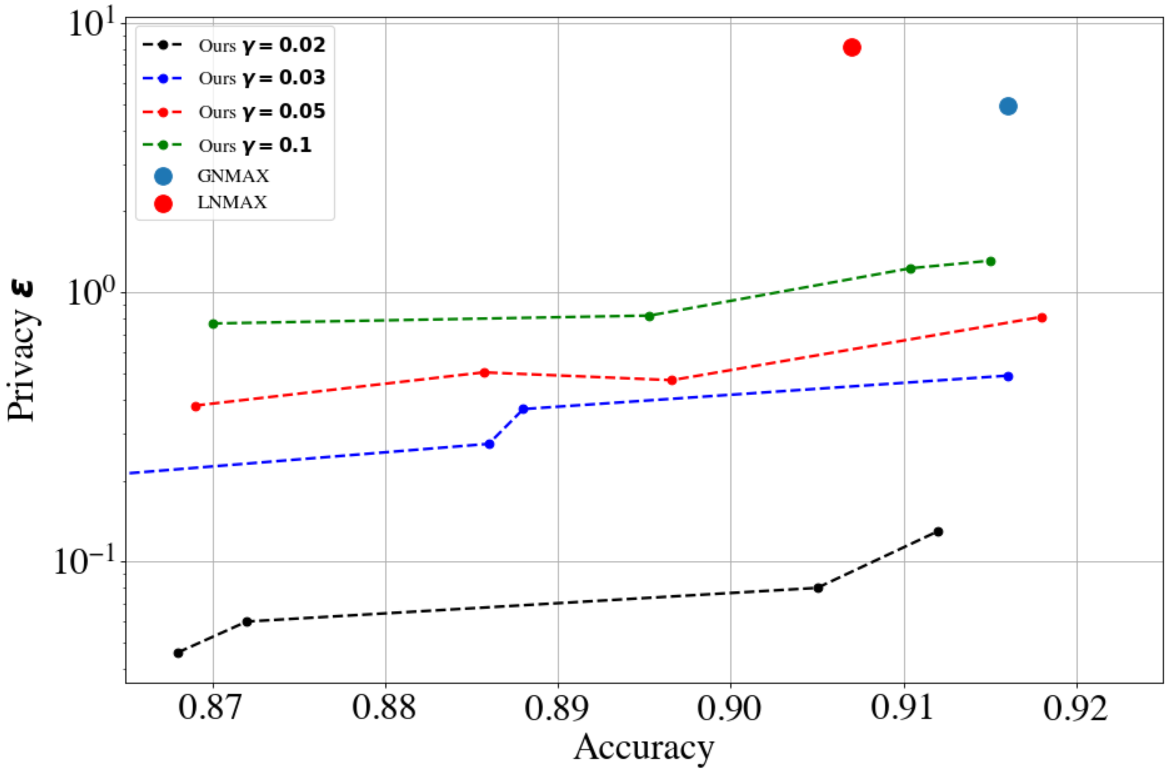


Figure 4.4: Tradeoff between utility and privacy for Private-kNN on SVHN. In this figure, different curve are generated with different sampling ratio γ. In each curve, we set different query number for student, and compute the total privacy and accuracy at test set. $\sigma_1 = 240, T = 480, \sigma_2 = 60, k = 800$. We also plot the results reported in PATE. It shows that the privacy cost of our model could achieve nearly two order of magnitude smaller privacy with better accuracy.

Then a student model is trained on the 735 answered queries with pseudo labels and VAT regularization, which achieves accuracy 98.8%. In Table 4.1, comparing to PATE of Laplace mechanism "LNMAX" and Gaussian mechanism "GNMAX", our method achieves significantly better accuracy-privacy trade-off. For instance, when we control the same number of queries between "GNMAX" and ours, Private-kNN achieves similar accuracy as 98.8% over 98.5%, but much better privacy cost as $\epsilon = 0.47$ compared to $\epsilon = 1.97$ of "GNMAX". More surprisingly, with a strict privacy cost of $\epsilon = 0.47$, our method shows only 0.4% deficit to the non-private model performance 99.2%.

SVHN: As shown in Table 4.2, we run our model for two iterations with hyper-parameters $k = 800, T = 480, \sigma_1 = 200, \sigma_2 = 60$ and $\gamma = 0.03$. In the first iteration, kNN with HoG feature provides 82.5% accuracy on 1022 answered queries. By retraining a CNN with the queried labels, it improves to 86.6%. In the second iteration, another 3000 queries are conducted via kNN, and 1917 queries are returned. KNN accuracy is evaluated on the selected queries, which passed the noisy screening check, whereas the retrain CNN is evaluated on the public testing set after self-supervised training and achieves 91.6% accuracy. These procedures can be iterated many times, where we empirically observe that two rounds can bring the converged performance. In total, we spend the privacy cost on 6000 samples for noisy screening and noisy aggregation with $2919(1022 + 1917)$ samples.

Table 4.1 shows the comparison to “GNMAX” and “LNMAX”. Both “GNMAX” and ours achieve better privacy accuracy trade-off than “LNMAX”. Though the number of queries in “LNMAX” is only 100, the privacy cost is as high as 8.19. This is mainly from the inefficiency of the Laplace mechanism compared to Gaussian mechanism, as Gaussian mechanism shows 30 times more queries with half of the privacy cost (4.96 over 8.19). Further comparing our method with “GNMAX”, with the similar number of queries and exactly the same accuracy, we achieved 0.49 privacy cost, which is significantly smaller than 4.96 from “GNMAX”. Notice that privacy cost below 1 indicates an excellent system which is ready for *practical* applications.

Figure 4.4 shows by varying sampling ratio γ, the privacy cost ϵ changes with respect to the number of queries. “GNMAX” and “LNMAX” are also compared. In the figure, all of our methods are advantageous, i.e. consistently lower privacy cost than those two spots of “GNMAX” and “LNMAX”. Further exploring different levels of γ, we observe that all the curves are mostly flat, which indicates that when pushing accuracy high, the increase of privacy cost is marginal. Moreover, it shows that with different sampling ratio,

Table 4.3: Real sensitive dataset evaluation on CelebA Liu et al. [2015] and Market1501 Zheng et al. [2015], we set $\tau = 10$ for both GNMAX and ours. T is the number of teachers in teacher ensemble model. We compare different methods under high privacy and low privacy regime. $\delta = 10^{-6}$ for CelebA and $\delta = 10^{-5}$ for Market.

Dataset	Methods	Parameter				#Queries	ϵ	Acc.	NP Acc.
		T	k	σ	γ				
CelebA	GNMAX	300	-	150	-	600	7.72	85.0%	89.5%
	GNMAX	800	-	300	-	500	3.31	84.4%	
	Ours	-	800	50	0.05	800	1.24	85.2%	
	Ours	-	800	100	0.10	800	1.20	84.9%	
Market1501	GNMAX	300	-	100	-	800	13.41	86.8%	92.1%
	GNMAX	300	-	250	-	80	1.41	85.6%	
	Ours	-	300	100	0.05	1200	0.67	88.8%	
	Ours	-	300	100	0.10	1200	1.38	89.2%	

our method can achieve different level of privacy cost. Across the large range of sampling ratio (0.02 to 0.1), we can push all the performance between 91% to 92%, which is at the same level of “GNMAX” and “LNMAX”, while with an order of magnitude lower privacy cost.

4.4.1 CIFAR-10 Evaluation

CIFAR-10 is a general objection classification task, where the PATE model is hard to apply as the data partitioning results in limited training data for each teacher model. For instance, each teacher model is assigned only 200 data if we partition the training set into 250 teachers, which is far from sufficient to train a deep neural network. For our experimental setting, we split total $60k$ data into three parts: $30k$ is treated as private data, $29k$ is for unlabeled public data, and $1k$ for testing.

Regarding this dataset, a competitive method, termed Noisy-SGD Abadi et al. [2016], achieved accuracy 70% and $\epsilon = 4$ when $\delta = 10^{-5}$ as shown in Table 4.1 CIFAR-10. In the Noisy-SGD setting, CIFAR-100 is leveraged to pre-train a model. For fair comparison, we also use the CIFAR-100 model as a pre-trained model for each teacher in PATE Papernot et al. [2018] and extract the initial feature with it for the Private-kNN. The latter iterative updating of the student model remains the same. For PATE performance, we notice that after model aggregation, it is below 50% even after we set $\epsilon \gg 10$.

In our implementation, with the initial CIFAR-100 extracted feature, the Private-kNN aggregator answers 3877 over total 18000 queries from the public domain. We set neighbor $K = 300, T = 210, \sigma_1 = 85, \sigma_2 = 20$, sampling ratio $p = 0.2$ and adopt the same model architecture as Abadi et al. [2016]. The model architecture contains three convolutional layers with $32, 64, 128$ filters in each convolution layer. The non-private baseline of this model reaches 80.5% accuracy when trained with $30k$ private data, whereas SOTA models present over 10% higher accuracy. The reason of not leveraging the SOTA models in this experiment is because, for fair comparison to Noisy-SGD, we aim to emphasize the privacy-utility trade-off, but not the best utility. Our method achieves an accuracy of 70.8% with privacy cost $\epsilon = 2.92$, which thoroughly outperforms Noisy-SGD.

Notice that the privacy cost in Noisy-SGD is spent on every parameter of the network; Thus, their retraining only involves the fully connected layers. Another difference is, we assume there exists unlabeled auxiliary data in public domain while Noisy-SGD Abadi et al. [2016] directly train a private model with $50k$ private data. Comparing to Noisy-SGD, our Private-kNN is indeed model agnostic, no restriction on network structure or optimization methods for retraining a student model, whereas clipping gradient in Noisy SGD may result in unstable optimization.

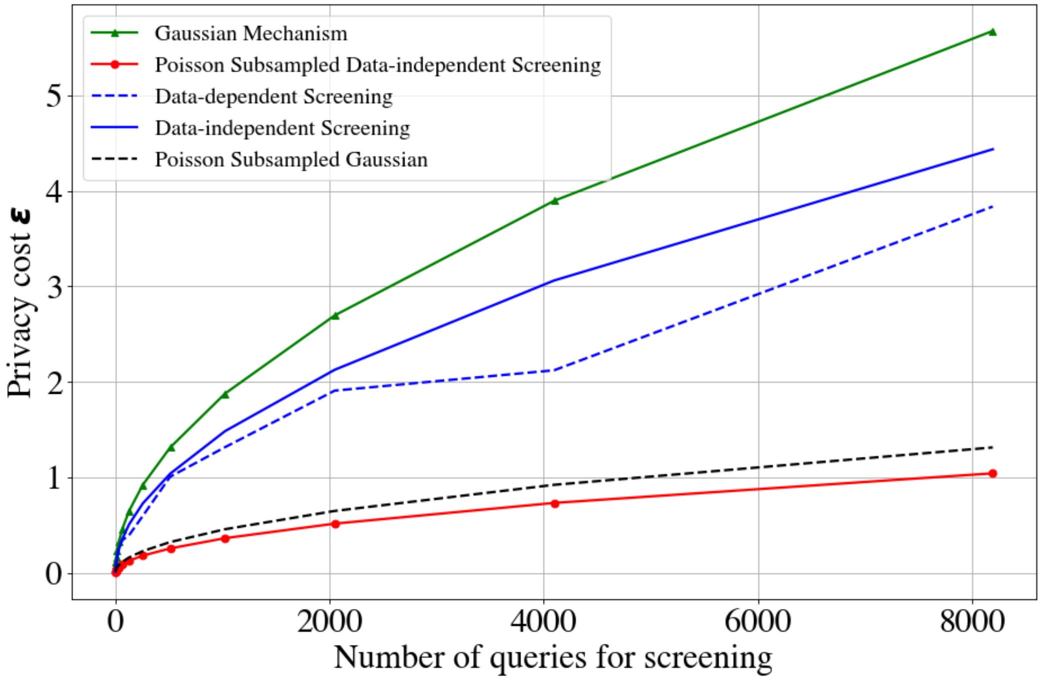


Figure 4.5: The privacy cost of answering 8192 queries with five randomized algorithms in the noisy screening process. The green line is the strong composition of Gaussian Mechanism used in PATE, the black dash line shows the privacy cost of Poisson subsampled Gaussian Mechanism after 8192 rounds' composition. The blue line is the ϵ of strong composition of data-independent screening and the blue-dash line is the strong composition of data-dependent screening. The red line is the Poisson subsampled data-independent screening $\mathcal{M}_s$. The sampling ratio $\gamma = 0.25$, $\sigma_1 = 85, k = 300$.

4.4.2 Noisy Screening for Less Privacy Cost

Screening and private voting are the core components of privacy guarantee. The purpose of screening is to filter out queries where there is no consensus among the votes. The privacy cost on screening is the major expense as reported in PATE Papernot et al. [2017] since we need to pay privacy cost for each query.

We investigate different private screening methods by exploring their privacy cost with respect to a different number of queries. In Figure 4.5, each screening algorithm is required to answer 8192 queries on the CIFAR-10 dataset, and the cumulative privacy cost is plotted along the y-axis. We use the HoG feature in the initial iteration for our

Private-kNN and set the sampling ratio of $\gamma = 0.25$. The noisy scale $\sigma_1 = 85$, threshold $T = 210$ and $k = 300$ is used for all screening methods.

The green line describes the privacy cost of Gaussian mechanism applied by PATEPapernot et al. [2018], serving as our baseline. It achieves $\epsilon = 5.67$ after privately screening 8192 queries. The black dash line demonstrates the privacy amplification by Poisson sampling Zhu and Wang [2019] with the same Gaussian mechanism. The privacy cost improves to 1.313. Even though, the original data-splitting setting in PATE prevents it to benefit from sub-sampling. The red line shows our data-independent screening method composed of Poisson sampling achieves $\epsilon = 1.04$. The blue line and the blue dash line show the result of incorporating our data-independent and data-dependent analysis of screening into PATE. It improves ϵ from 5.67 to 4.43 with the data-independent screening and 3.83 with the data-dependent screening.

When compared to the black-dash line (sub-sampled Gaussian), our method saves 26% privacy budget with the same screening results. Our method allows to answer more queries from the public domain, which is of essential importance, especially when the training task itself is tough. For example, employing self-supervised training with CIFAR-10 requires at least 4000 ground-truth labeled data Xie et al. [2019]. Then, the minimum number of queries demands at least 10000, since empirically, more than 50% data fails to pass the screening check. Our advantageous privacy cost 1.04 makes it a practical solution for private training with more difficult machine learning tasks.

4.4.3 Real Private Datasets Evaluation

We show that our Private-kNN is a practical framework that indeed can apply to real private datasets, i.e., face attribute classification from CelebA Liu et al. [2015] and body attribute classification from Market1501 Zheng et al. [2015]. We aim to develop an attribute classification model, where the adversary is hard to detect whether one particular image has been used in training set with high probability. Both of the datasets target the human or face related tasks, where identity is crucial privacy to be preserved. Notice that they are multi-label classification tasks other than binary classification, which are more challenging. To reduce the privacy budget of multi-label tasks, we apply a τ approximation method where the basic idea is that, each neighbor could at most vote for τ attributes, or their total votes will be clipped to τ. The detailed definition and privacy guarantee can be found in Appendix. In our setting, we do not conduct noisy screening for multi-label classification because it is hard to guarantee all the labels within one query pass the screen.

CelebA is a large-scale face attribute dataset with more than $220k$ celebrity images, each with 40 attribute annotations. According to data splitting, we take the $160k$ training data as private data. From the $60k$ testing data, depending on the volume to be queried, i.e. 600 queries, the rest 59400 images are automatically regarded as testing. The non-private baseline is 89.5% trained via a Resnet50m structure. We apply PATE as another baseline. Since each image have 40 attributes, the global sensitivity grows as large as the dimension of attributes. We apply τ-approximation method to limit the range of global sensitivity and also consider the trade-off induced by the different τ. In Table 4.3, by choosing the parameters, when the privacy cost is smaller than “GNMAX”, we achieve clear better accuracy of 85.18% compared to 84.4%. When the accuracy is at the same level around 85%, our method achieves significantly lower privacy cost 1.20 compared to 7.72 of “GNMAX”.

Market1501 contains 1501 identities and 32668 images, where each image has 30 attributes. We split original training set for private data and validation set as unlabeled public data, performance is evaluated on original testing set. In this task, data-splitting is stressful. The total private data contains only 750 identities. For PATE, to guarantee the teacher models' independence, we need to partition the private data with respect to the identities. A meaningful privacy cost requires sufficient many teacher models, i.e., $K = 300$. With such many partitions of the private data, each teacher is trained with around 40 images from 2 identities, and the non-private accuracy of each teacher is only 71%.

Shown in Table 4.3, our method is able to answer 1200 queries compared to 80 in "GNMAX" where two methods achieve similar privacy cost 1.414 and 1.377. The significantly more queries lead to performance boost as 89.18% compared to GNMAX 85.61%. To push up the performance for "GNMAX"(i.e., from 85.6% to 86.8%), we tune the privacy-utility trade-off and the privacy cost goes high up to 13.41, which prevents the trade-off from improving performance further. We provide a relative close trade-off, accuracy 88.8%, and privacy $\epsilon = 0.67$, both of which are far better than the "GNMAX". The detailed utility and privacy trade-off can be found in the appendix, which demonstrates the consistent advantages of our method in real private tasks.

4.5 Conclusion

In this work, we propose a data-efficient privately releasing of k nearest neighbor framework, termed Private kNN, to overcome the limited private data issue in vision applications. A new Renyi-dfferential privacy analysis for noisy screening procedure is proposed, which allows our model to answer 10 times more queries compared to other private knowledge transfer models such as PATE. Extensive experiments are conducted across five vision benchmarks, showing that our method achieves comparable or better accuracy than PATE while saving more than 90% privacy cost. Specifically, the two realistic identity related classification tasks demonstrate that our private kNN achieves high utility with practical DP guarantees.

Chapter 5

Ind-KNN: Private kernelized nearest neighbors with Individual Rényi filter

5.1 Introduction

Differential privacy (DP; Dwork et al. [2006, 2014b]) is a promising approach for mitigating privacy risks in machine learning (ML). The predominant setting for private ML is to produce the model learned from sensitive data using DP primitives, a.k.a. private training [Chaudhuri et al., 2011, Kasiviswanathan et al., 2011, Abadi et al., 2016]. The resulting trained model can then be safely deployed with peace of mind, because DP ensures that no individual training sample can be identified from the model itself or its downstream predictions.

Unfortunately, private training comes with several irky properties that hamper its real-life deployment. To begin, private training comes at a significant computation cost that can be restrictive in many applications. The NoisySGD algorithm [Abadi et al., 2016] requires per-sample gradient computation, which is much more computation- and memory-intensive than standard training.

Secondly, private training outputs a static model that cannot easily adapt to a changing dataset. For instance, additional data can arrive in a streaming fashion continuously. Also, training data could be mislabeled or corrupted [Chen et al., 2017, Jagielski et al., 2018] and the model needs to be patched accordingly. In addition, if the model is trained on user data, privacy regulations such as GDPR entitle the user to request the removal of their data from the model [Ginart et al., 2019, Guo et al., 2020, Bourtoule et al., 2021] with the so-called *right to be forgotten* [Mantelero, 2013]. These requirements can be satisfied by periodically re-training the model, but such an approach is not applicable to private training due to its high computation cost as well as privacy degradation after repeated training runs.

Thirdly, privacy training operates under a very strong threat model in which all downstream users can collude with each other in a coordinated attack on any individual training sample. Sometimes it makes sense to make realistic assumptions that limit the adversaries' information or resources. For example, Harvard's Privacy Tools project (now OpenDP) adopts a weaker threat model where each downstream user keeps the results to themselves [Gaboardi et al., 2016]. In this way, they each get to spend the privacy budget independently of everyone else and enjoy higher utility. Private training unfortunately does not have a means to benefit from having weaker adversaries.

To address these issues of private training, we revisit a viable but less-known alternative setting in differentially private machine learning known as *privacy-preserving prediction* (or simply *private prediction*) [Dwork and Feldman, 2018]. Instead of privately training the models and then using the model for predictions, private prediction aims at generating a sequence of predictions using the data directly. Notable methods include those that perturb the predictions of non-private models [Dwork and Feldman,

Table 5.1: The amortized computational and privacy cost of answering $T = 2000$ queries on CIFAR-10. The median accuracy of all approaches across five independent runs is aligned to 96.0%. We estimate the amortized computational cost by calculating the averaged time spent (in seconds) to answer a single query, which is the total time of training divided by T in Linear NoisySGD [Feldman and Zrnic, 2021] and the total time of predictions divided by T in Private kNN [Zhu et al., 2020] and Ind-KNN. We use $\delta = 10^{-5}$. In the retraining scenarios, we assume that a retraining request is made every answering 100 queries, resulting in a total of 20 retraining requests among T queries.

	NoisySGD	NoisySGD (with retrain)	Private kNN	Ind-KNN (ours)	Ind-KNN+hashing (ours)
Computational cost (s)	0.008	0.16	0.12	0.25	0.04
Privacy loss (ϵ)	1.5	6.2	4.1	2.0	3.2

2018, Papernot et al., 2018, Bassily et al., 2018, Dagan and Feldman, 2020], or those that perturb the voting scores of the nearest neighbors [Zhu et al., 2020]. These methods require no changes to the (non-private) data workflow, and thus could more easily adapt to changing data.

From the privacy-utility trade-off point of view, the private prediction setting may appear to be counter-intuitive, because, for every prediction that it generates, a unit of privacy budget is spent. It is unreasonable to expect private prediction methods to outperform private training methods such as NoisySGD when we need to make many predictions. This was well-documented in the work of van der Maaten and Hannun [2020]. However, in the aforementioned situations when either frequent data updates are needed or a weaker adversary is assumed[1], private prediction methods can significantly outperform NoisySGD (See Table 5.1 and Figure 5.3 for an illustration). In fact, we will demonstrate that when combined with modern DP accounting techniques, data-adaptive DP algorithm design, and some clever reuse of previous predictions, a small privacy budget can answer thousands of queries without significantly increasing the privacy loss.

In this work, we propose *Individual Kernelized Nearest Neighbors* (Ind-KNN) — a new private prediction mechanism that significantly increases the number of queries one

[1] Consider the example of a recommendation system, each user makes a much smaller number of predictions than all users collectively.

can answer with an individualized Rényi Differential Privacy accountant and other techniques. Intuitively, in KNN prediction, training samples that do not belong to the query's neighbor set do not contribute to the prediction, and hence their privacy cost should be negligible. We show that by slightly modifying KNN and leveraging Rényi filter [Feldman and Zrnic, 2021] to account for the privacy cost of each sample individually, we can realize this intuition in the privacy accounting and allow each training sample to participate in the query response until its own privacy budget is exhausted. In effect, common queries can be answered with relatively low privacy costs due to a large number of similar samples present in the training set.

Experimental results. We summarize our experimental results as follows:

1. We show that Ind-KNN consistently outperforms the private prediction benchmark, Private-kNN [Zhu et al., 2020], across four vision and language tasks for a range of epsilon between $[0.5, 2.0]$.

2. We demonstrate that Ind-KNN is a viable alternative to private training methods even in a static data setting. Our results indicate that Ind-KNN achieves higher accuracy than NoisySGD when answering less than 2000 queries on CIFAR-10 under $(1.0, 10^{-5})$-DP.

3. For frequent data updates, Ind-KNN significantly outperforms the private training benchmark Linear NoisySGD [Feldman and Zrnic, 2021]. As shown in Table 5.1, Linear NoisySGD requires a DP budget of $\epsilon = 6.2$ to achieve an accuracy of 96.0% on 2000 queries of CIFAR-10, while Ind-KNN only requires $\epsilon = 2.0$.

4. We describe two simple techniques that significantly enhance the computational efficiency and utility of Ind-KNN. First, we show that incorporating hashing tricks into Ind-KNN can provide a 6× speedup in making predictions, with only a neg-

ligible drop in accuracy. Additionally, we propose to reuse the results of previous queries via post-processing, which allows Ind-KNN to answer an additional 1000 queries on CIFAR-10 without compromising in privacy or utility.

Related work and novelty. The problem of private prediction was pioneered by Dwork and Feldman [2018] as a weakened goal for private machine learning. Model-based approaches for private predictions either require analyzing the stability of model training [Dwork and Feldman, 2018, Dagan and Feldman, 2020] or to enforce stability of prediction via subsample-and-aggregate [Papernot et al., 2018, Bassily et al., 2018]. Our method is closest to Private kNN [Zhu et al., 2020] but uses kernel-weighted neighbors with a variable K instead of a fixed K. This change is critical for adapting the individual Rényi DP accountant (and filter) for our purpose. Other components such as adaptive noise-level, prediction reuse, and the fast hashing trick are new to this paper. Technically, we apply the same individual Rényi filter [Feldman and Zrnic, 2021] that retires data samples when their privacy budget runs out. The difference is that we applied it to KNN rather than noisy gradient descent. KNN naturally has bounded support thus it is efficient to maintain the individual RDP accountants.

5.2 Preliminaries

We start with the definition of privacy-preserving prediciton.

Privacy-preserving prediction. We now formally state the setting of privacy-preserving prediction. Consider a prediction task over a domain $\mathcal{X}$ and label space $\mathcal{Y}$. The prediction interface $\mathcal{A}$ has access to a private dataset $S = (x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, which outputs a value $a \in \mathcal{Y}$ if given a query $q \in \mathcal{X}$. We denote by Q a query generating algorithm that can adaptively generate a query given the previous released outputs. Namely, we denote by $\mathcal{A}(S) \rightleftharpoons_T Q = (q_t, a_t)_{t=1}^T$ the sequence of query-response pairs generated by the prediction interface $\mathcal{A}$ over a sequence of length T queries on dataset S, where $a_t = \mathcal{A}_t(a_1, ..., a_{t-1}, S, q_t)$.

The privacy guarantee of private prediction is applied for a sequence of predictions generated by the interface $\mathcal{A}$.

Definition 5.2.1 (Privacy-preserving prediction interface). [Dwork and Feldman, 2018] A prediction interface $\mathcal{A}$ is (ϵ, δ)-differentially private, if for every interactive query generating algorithm Q, the output $\mathcal{A}(S) \rightleftharpoons_T Q = (q_t, a_t)_{t=1}^T$ is (ϵ, δ)-DP with respect to dataset S.

Privacy-preserving prediction algorithms can be useful in a variety of situations where releasing a DP model is restricted or not practical. For example, companies that train a privacy-preserving model and only require making a limited number of predictions can rely on a prediction interface instead of releasing the entire model. In addition, in health or financial data scenarios, private prediction algorithms allow for a cloud-based interface to be exposed, which can also help to ensure compliance with regulatory requirements.

Individual RDP. Our privacy analysis relies on individual privacy loss, which accounts for the maximum possible impact of an individual data point on a dataset. The following definition states the individual privacy loss in terms of Rényi divergence.

Definition 5.2.2 (Individual RDP [Feldman and Zrnic, 2021]). Fix $n \in \mathcal{N}$ and a private data point $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$. We say that a randomized algorithm $\mathcal{A}$ satisfies (α, ρ)-individual Rényi differential privacy for z if for all datasets $S = (z_1, ..., z_m)$ such that $m \leq n$ and $z_i = z$ for some i, it holds that

$$D_\alpha^{\leftrightarrow}\big(\mathcal{A}(S)||\mathcal{A}(S^{-i})\big) \leq \rho,$$

where $D_\alpha^{\leftrightarrow}$ denotes the max of $D_\alpha\big(\mathcal{A}(S)\|\mathcal{A}(S^{-i})\big)$ and $D_\alpha\big(\mathcal{A}(S^{-i})\|\mathcal{A}(S)\big)$.

Note that the individual RDP parameter ρ is a function of a data point z, and thus does not imply the standard RDP guarantee in Definition 1.3.1. However, we can obtain the standard RDP guarantee by requiring that all data points z satisfy individual RDP with the same ρ.

Now, we present an example of individual RDP computation on Gaussian mechanism.

Lemma 5.2.3 (Linear queries with Gaussian mechanism [Feldman and Zrnic, 2021])**.** *Let $S = (z_1, ..., z_n) \in (\mathcal{X} \times \mathcal{Y})^n$. Suppose that $\mathcal{A}$ is a d-dimensional linear query with Gaussian noise addition, $\mathcal{A}(S) = \sum_{j\in[n]} q(z_j) + \mathcal{N}(0, \sigma^2\mathbf{1}_d)$ for some $q : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}^d$. Then $\mathcal{A}$ satisfies*

$$D_\alpha^{\leftrightarrow}\big(\mathcal{A}(S)||\mathcal{A}(S^{-i})\big) \leq \frac{\alpha||q(z_i)||_2^2}{2\sigma^2}$$

individual RDP for z_i. Note that by replacing$||q(z_i)||_2$ with the ℓ_2 global sensitivity of $q(\cdot)$, the expression above recovers the standard RDP of Gaussian mechanism.

The following theorem states the composition property of individual privacy. For a sequence of algorithms, as long as the composition of individual RDP parameters does not exceed a pre-specified budget for all data points, the output of the adaptive composition preserves the standard RDP guarantee.

Theorem 5.2.4 (Corollary 3.3 [Feldman and Zrnic, 2021])**.** *Fix any $G \geq 0$ and any $\alpha \geq 1$. For any input dataset $S = (z_1, ..., z_n)$ and for any sequence of algorithms $\mathcal{A}_1, ..., \mathcal{A}_T$, let $\rho_t^{(i)}$ denote the individual RDP parameter of the t-th adaptively composed algorithm $\mathcal{A}_t$ with respect to z_i. if $\sum_{t=1}^T \rho_t^{(i)} \leq G$ holds almost surely for all $i \in [n]$ then the adaptive composition $\mathcal{A}^{(T)}$ satisfies (α, G)-RDP.*

The composition rule described above is known as fully adaptive composition [Rogers et al., 2016], which takes *adaptively-chosen* privacy parameters instead of pre-specified ones in the classical adaptive composition. This type of composition is necessary for individual privacy since the individual RDP parameters themselves are random variables that depend on the outputs released by previous composed mechanisms.

To implement the composition above, we need a tool called *Rényi filter*, which is designed to ensure that the composed individual privacy parameters is maintained within a given budget G for all individuals. In practice, we can implement Rényi filter by providing each data point with an individual accountant that estimates its composed individual RDP $\sum_{t=1}^{T} \rho_t^{(i)}$ and dropping the data point once it exceeds the budget, as shown in Algorithm 2.

However, despite its tighter privacy analysis, this technique has been criticized for its computational cost of tracking individual privacy costs for all data samples. In this work, we demonstrate that KNN works seamlessly with the individual RDP accountant. Only selected neighbors are required to update their individual privacy accountants, which significantly reduces the computational cost.

Algorithm 2 Adaptive composition $\mathcal{A}^{(T)}$ with Rényi filter

```
**Input**: Dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, sequence of algorithms $\mathcal{A}_{1:T}$ and privacy budget $G$.
**for** $t = 1, ..., T$ **do**
    For all $z_i \in S$, compute
    $\rho_t^{(i)} = \sup_{S' \in \mathcal{S}} D_\alpha^{\leftrightarrow} \left(\mathcal{A}(a_{1:t-1}, S') || \mathcal{A}(a_{1:t-1}, S'^{-i})\right)$
    Update the active set $S = \{z_i | \sum_{j=1}^{t} \rho_j^{(i)} \leq G\}$
    Compute $a_t = \mathcal{A}_t(a_{1:t-1}, S)$
**end for**
**Return** $(a_1, ..., a_T)$
```

5.3 Private Prediction with Ind-KNN

To overcome the limitations of private training, we propose *Individual Kernelized Nearest Neighbor* (Ind-KNN)—a k-nearest neighbor-based private prediction algorithm that achieves a comparable DP guarantee and test accuracy to that of private training.

Notations and setup. We focus on the task of multi-class classification. Given a private dataset $S = (x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})$, we assume y_i is an one-hot vector over c class, i.e., $y_i \in \{0, 1\}^c$. Let $\phi(\cdot)$ denote a *public* feature extractor that maps the input $x \in \mathcal{X}$ to a fixed-length feature representation $\phi(x) \in \mathcal{R}^d$. This could be image features extracted from the penultimate layer of a ResNet50 pre-trained model or language features extracted from the final layer of a transformer model. The feature extractor is used to encode both the private dataset and public queries.

Algorithm 3 Privacy-preserving prediction with naive kNN

```
Input: Dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, sequence of queries $q_1, ..., q_T$, number of neighbor $k$
    and the noisy scale $\sigma$.
for $t = 1$ to $T$ do
    $\mathcal{N}_k$ := top k nearest neighbors of the query $q_t$
    $a_t = \arg\max_{j \in [c]} \left(\sum_{i \in \mathcal{N}_k} y_i + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{1}_c)\right)_j$
end for
Return $(a_1, ..., a_T)$
```

Previously, k-Nearest Neighbor (kNN) has been used for privacy-preserving prediction by Zhu et al. [2020] (Algorithm 3). In this method, when a query q_t arrives, the top k nearest neighbors are selected from the private dataset based on the distance in the feature space, and their labels are utilized for prediction through a Gaussian mechanism.

However, the privacy loss of Algorithm 3 accumulates rapidly as the number of queries increases, owing to its conservative privacy analysis that bounds the worst-case individual privacy loss over all individuals. In contrast, the Ind-KNN approach emphasizes individual privacy accounting, providing precise control over privacy loss at an individual data level. This allows each data point's privacy to be charged by the exact amount of its contribution to the query response, and private data is removed once its own privacy budget has been exhausted.

We propose a novel solution *Individualized Kernelized Nearest Neighbor* (Ind-KNN)

Algorithm 4 Kernelized-nearest-neighbor with individual privacy accounting (Ind-KNN)

```
Input: Dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, the kernel function $\kappa(\cdot,\cdot)$, the threshold $\tau$, sequence of queries $q_{1:T}$, the noisy scale $\sigma_1$, $\sigma_2$ and the individual budget $B$.
Initialize individual budget $z_i = B, \forall i \in [n]$.
for $t = 1$ to $T$ do
    Update the active set $S = \{(x_i, y_i) | z_i \geq \frac{1}{2\sigma_1^2}\}$.
    Release the number of selected neighbors: $K_t := \sum_{(x_i,y_i)\in S} \mathbb{I}[\kappa(x_i, q_t) \geq \tau] + \mathcal{N}(0, \sigma_1^2)$.
    for $(x_i, y_i) \in S$ do
        Update the remaining budget $z_i$ after releasing $K_t$: $z_i = z_i - \frac{1}{2\sigma_1^2} \cdot \mathbb{I}[\kappa(x_i, q_t) \geq \tau]$.
        Evaluate individual contribution $f_t : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}^c$ as $f_t(x_i, y_i) := \min\big(\kappa(x_i, q_t) \cdot y_i \cdot \mathbb{I}[\kappa(x_i, q_t) \geq \tau], \sigma_2\sqrt{2K_t \cdot z_i} \cdot \mathbf{1}_c\big)$
        Update the remaining budget $z_i$ after releasing label: $z_i = z_i - \frac{||f_t(x_i,y_i)||_2^2}{2\sigma_2^2 \cdot K_t}$.
    end for
    $a_t = \arg\max_{j\in[c]} \big(\sum_{(x_i,y_i)\in S} f_t(x_i, y_i) + \mathcal{N}(\mathbf{0}, \sigma_2^2 \cdot K_t \cdot \mathbf{1}_c)\big)_j$.
end for
Return $(a_1, ..., a_T)$
```

in Algorithm 4. Intuitively, nearest neighbor-based prediction leak little to no private information when the query point is near a dense region of the training data. This is because the result of the query is determined by a large number of training samples and hence is insensitive to individual training points. We make several modifications to Private kNN to realize this intuition.

First, we introduce individual privacy accounting by assigning each private data point (x_i, y_i) with a pre-determined privacy budget B, represented by the variable $z_i := B$. For each query, the algorithm updates the private dataset S to only include data points where $z_i > \frac{1}{2\sigma_1^2}$. This ensures that the privacy budget for each individual is not exceeded.

Second, Ind-KNN improves upon Algorithm 3 by utilizing a pre-specified threshold τ and a kernel-based similarity function $\kappa(\cdot,\cdot)$ to select only neighbors with similarity above τ. This approach allows only the selected neighbors to be accountable for their privacy loss, preserving the privacy budget of un-selected private individuals for future queries.

It is worth noting that simply selecting the exact top k neighbors, as in Algorithm 3, is not consistent with individual privacy loss. This is because the decision of selection is dependent on the dataset: a $k+1$th nearest neighbor in one dataset may be the top nearest neighbor in another dataset. Hence, all private data points must account for their individual privacy loss, even if only a subset of them contribute to the prediction, according to the definition of individual RDP (Definition 5.2.2).

Moreover, Ind-KNN employs kernel weights for prediction aggregation instead of equal weight for all nearest neighbors. In our experiments, we consider two types of kernel functions, RBF and cosine, to measure the similarity. For example, the RBF kernel is defined as $\kappa(x, q_t) := e^{\frac{-||\phi(x)-\phi(q_t)||_2^2}{\nu^2}}$, where $\phi(x)$ and $\phi(q_t)$ are the encoded feature and ν is a scalar parameter. This adaptation, made possible by individual privacy accounting, results in a more accurate characterization of each individual's contribution to the query. However, changing from equal weight to kernel weight in Algorithm 3 would not alter its privacy analysis (as the worst-case kernel weight is bound by 1), but would instead decrease the signal-to-noise ratio (each neighbor's contribution would be less than 1).

Finally, Ind-KNN dynamically adjusts the magnitude of noise added to the noisy prediction by publishing the number of neighbors at each query. We find that adding noise with variance proportional to K_t is crucial for good performance. This allows us to adjust the margin of the voting space — the difference between the largest and the second largest coordinate of $\sum_{(x_i,y_i)\in S} f(x_i, y_i)$ adapted to the noise scale. Specifically, when the margin is significant, adding larger noise will not change the output label, but it reduces each individual's individual privacy loss proportional to the reciprocal of K_t, enabling them to participate in more queries in the future.

Algorithm. The modifications made in Ind-KNN are summarized in Algorithm 4. Specifically, since the number of selected neighbors is considered private information, each selected neighbor accounts for its individual privacy loss due to releasing $\mathbb{I}[\kappa(x_i, q_t) \geq \tau]$

by subtracting z_i with $\frac{1}{2\sigma_1^2}$ at line 7 of Algorithm 4. Meanwhile, $f_t(x_i, y_i)$ at line 8 accounts for the individual contribution of releasing its label associated with kernel weight. The first term represents the "weighted" one-hot label for selected neighbors and all-zero vectors for unselected private data. The second term $\sigma_2\sqrt{2K_t z_i}$ ensures that the incurred individual privacy loss of releasing label will not go beyond the remaining budget z_i.

Lemma 5.3.1 (Individual RDP of releasing a_t)**.** *Given a query q_t, for each (x_i, y_i), define the function $f_t : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}^c$ as line 8 in Algorithm 4. Then the release of* $a_t = \arg\max_{j\in[c]} \big(\sum_{(x_i,y_i)\in S} f_t(x_i, y_i) + \mathcal{N}(0, \sigma_2^2 \cdot K_t \cdot \mathbf{I}_c)\big)_j$ *satisfies* $(\alpha, \frac{\alpha\cdot||f_t(x_i,y_i)||_2^2}{2\sigma_2^2\cdot K_t})$ *individual RDP for each (x_i, y_i).*

The proof directly follows from Lemma 5.2.3 and the post-processing property of individual privacy. Note that for unselected private data, their individual privacy loss is always zero since their individual contribution $f(\cdot)$ is zero.

Theorem 5.3.2. *Algorithm 4 satisfies $(\alpha, B\alpha)$-RDP for all $\alpha \geq 1$.*

Proof: The proof makes use of the facts that: (1) the decision rule for "being selected" is not influenced by any other private data points, thus, "unselected" neighbors does not incur any individual privacy loss. (2) adding/removing one selected neighbor would only change $\sum_{(x_i,y_i)\in S} \mathbb{I}[\kappa(x_i, q_t) \geq \tau]$ by one, thus the release of K_t satisfies $(\alpha, \frac{\alpha}{2\sigma_1^2})$ individual RDP for selected neighbors. (3) the release of label associated with the kernel weight satisfies $(\alpha, \frac{\alpha||f_t(x_i,y_i)||_2^2}{2\sigma_2^2 K_t})$ individual RDP.

The privacy analysis relies on individual RDP (Definition 2.4), which quantifies the maximum impact of adding or deleting a specific individual from any potential dataset to the prediction outcome, measured in terms of Rényi divergence.

We first demonstrate that only the selected neighbors have to account for their individual privacy loss. The decision rule for "being selected" is based on a comparison between the kernel weight and a data-independent threshold τ, which is not influenced by any other private data points. Therefore, "unselected" neighbors do not incur any individual privacy loss.

For each selected neighbor (x_i, y_i) at time t, its individual privacy analysis is broken down into two parts: the first part is the release of the number of neighbors $|\mathcal{N}_t|$, and the second part is the release of its label associated with the kernel weight.

Note that adding or removing one selected neighbor would only change $|\mathcal{N}_t|$ by 1, thus the individual RDP of releasing $|\mathcal{N}_t|$ at order α satisfies $\frac{\alpha}{2\sigma_1^2}$-RDP for all selected data. We next analyze the individual RDP of releasing the label. Fix a selected neighbor $z = (x_i, y_i)$, for all possible set of selected neighbors $\mathcal{N}_t = (z_1, ..., z_m)$ that include z, it holds that

$$D_\alpha^{\leftrightarrow}\bigg(\big(\sum_{j \in \mathcal{N}_t} \kappa(x_j, q_t) \cdot y_j \big) + \mathcal{N}(0, \sigma_2^2 K_t \mathbb{I}_c) || \big(\sum_{j \in \mathcal{N}_t \backslash z} \kappa(x_j, q_t) \cdot y_j \big) + \mathcal{N}(0, \sigma_2^2 K_t \mathbb{I}_c) \bigg) \leq \frac{\kappa(x_i, q_t)^2 \alpha}{2\sigma_2^2 K_t}$$

by the definition of individual RDP.

Finally, The "delete" step in the algorithm ensures that the privacy loss for each private data point (x_i, y_i) is bounded by a fixed value B, i.e., $\sum_{j=1}^t \Big((g_i + \frac{1}{2\sigma_1^2 \cdot K_j}) \cdot \mathbb{I}[(x_i, y_i) \in \mathcal{N}_j] \Big) \leq B$. According to the fully adaptive composition theorem of individual RDP (Theorem 2.5), by ensuring the sum is less than or equal to B for all time steps t and for all data point (x_i, y_i), the algorithm is shown to be $(\alpha, \alpha \cdot B)$-RDP. ■

Remark 5.3.3. We remark that the privacy guarantee of Ind-KNN is determined by the given individual budget, and remains the same regardless of the number of predictions made. However, as the number of predictions increase, the exclusion of private data may result in a degradation of the algorithm's utility.

Efficient Ind-KNN

In this section, we present two novel techniques that aim to improve the efficiency of Ind-KNN in terms of both utility and computational cost.

Ind-KNN with prediction reuse. The first technique improves the utility of Ind-KNN by exploiting the previously released predictions. We acknowledge that the query-response pairs that have been disclosed can be considered public information. Therefore, we incorporate those predictions into the active set S without any limitation on their privacy budgets. The results of our experiments demonstrate that this extension effectively mitigates the utility loss caused by the exclusion of private data points and improves the test accuracy when handling a large number of queries.

Ind-KNN with hashing. Algorithm 4 requires searching through all private data to answer each query, which can be computationally expensive if the private dataset is large. To address this issue, we present a variant of Ind-KNN that incorporates locality-sensitive hashing (LSH) [Gionis et al., 1999] for efficient nearest neighbor search. The full algorithm of Ind-KNN-Hash is in Algorithm Algorithm 5 LSH is a well-established technique to speed up the approximate nearest neighbor search. The principle behind the algorithm is to apply LSH to group private data points into "buckets" based on their hash values. When a query is made, the algorithm only needs to search the bucket that the query falls into, rather than searching through the entire dataset.

Concretely, Ind-KNN-Hash creates L hash tables $\mathcal{F} = (f_1, ..., f_L)$ with each of them maps a feature $\mu \in \mathcal{R}^d$ to a b-dimension bucket. For each table f, the algorithm generates b independent random Gaussian vectors from $\mathcal{N}(\mathbf{0}, \mathbf{1}_d)$, denoted by r_j for $1 \leq j \leq b$. Then we encode μ with $f(\mu) = (h_1(\mu), ..., h_b(\mu))$, where $h_j(\mu) = 0$ if $r_j^\top \mu < 0$, otherwise $h_j(\mu) = 1$. The algorithm then indexes all private data points into the hash tables using their encoded features. When a query q_t is received, the algorithm uses LSH to retrieve

Algorithm 5 Ind-KNN-Hash

```
**Input**: Dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, number of hash tables $L$ and the width parameter $b$, the kernel function $\kappa(\cdot,\cdot)$, the minimum kernel weight threshold $\tau$, sequence of queries $q_1, ..., q_T$, the noisy scale $\sigma_1$ and $\sigma_2$ and the individual budget $B$.
Initialize individual budget $z_i = B, \forall i \in [n]$.
Construct a LSH family: $\mathcal{F} = (f_1, ..., f_L)$, where $f_\ell : \mathcal{R}^d \to \{0,1\}^b$.
**for** $t = 1$ to $T$ **do**
    Retrieve the hash set: $\mathcal{F}(q_t)$.
    Update the active set $S = \{(x_i, y_i) | z_i > 0, (x_i, y_i) \in \mathcal{F}(q_t)\}$.
    The selected neighbors: $\mathcal{N}_t := \{(x_i, y_i) | \kappa(x_i, q_t) \geq \tau$ for all $i \in S\}$.
    Drop $(x_i, y_i)$ from $\mathcal{N}_t$ if $z_i \leq \frac{1}{2\sigma_1^2}$.
    Release $|\mathcal{N}_t|$: $K_t := |\mathcal{N}_t| + \mathcal{N}(0, \sigma_1^2)$.
    **for** $(x_i, y_i) \in \mathcal{N}_t$ **do**
        Update $z_i$ after releasing $K_t$: $z_i = z_i - \frac{1}{2\sigma_1^2}$.
        Evaluate individual "contribution": $g_i = \min\left(\frac{\kappa(x_i,q_t)^2}{2\sigma_2^2 \cdot K_t}, \sigma_2\sqrt{2K_t z_i}\right)$.
        Update $z_i$ after releasing label: $z_i = z_i - g_i$.
    **end for**
    Compute $a_t = \arg\max_{j \in [c]} \left(\sum_{i \in \mathcal{N}_t} \kappa(x_i, q_t) \cdot y_i + \mathcal{N}(\mathbf{0}, \sigma_2^2 \cdot K_t \mathbf{1}_c)\right)_j$.
**end for**
**Return** $(a_1, ..., a_T)$
```

a set of private data points that are hashed into the same bucket in at least one table, which is denoted by $\mathcal{F}(q_t)$. Finally, Algorithm 4 is called to label each query with a slight modification on the active set, which is now restricted to the retrieved data points with non-negative individual budgets. Typically, increasing the number of hash tables L and reducing the bucket size b results in more accurate neighbors but higher computational costs.

Incorporating LSH into Ind-KNN does not impose any additional privacy cost. This is because the encoding of each private data point is based on random Gaussian vectors and is executed independently of any other private data points.

5.4 Experiments

We consider the following standard image classification and language classification datasets. For each dataset, we take the training set as the private domain and the testing set as the public domain.

Image classification. We evaluate our method on two widely used image classification benchmarks, CIFAR-10 [Krizhevsky et al., 2009] and Fashion MNIST [Xiao et al., 2017]. For CIFAR-10, we employed the recent Vision Transformer (ViT) model [Dosovitskiy et al., 2020], which is pre-trained on the ImageNet-21k consisting of 14 million images and 21843 classes). The extracted from the ViT model are represented as 768-dimensional vectors. For Fashion MNIST, we consider the publicly available ImageNet-pretrained ResNet50 He et al. [2016] from Pytorch as the feature extractor. The model returns a 1000-dim vector for each input image.

Text classification. We utilize AG News [Zhang et al., 2015] and DBPedia [Lehmann et al., 2015] datasets to evaluate the performance of Ind-KNN on text classification tasks. We employ sentence embedding models [Zhao et al., 2022, Reimers and Gurevych, 2019] to extract features. Specifically, we utilize the `all-roberta-large-v1` sentence-transformer, which has been fine-tuned on a 1B sentence pairs dataset using a self-supervised contrastive learning objective. The extracted features are 1024-dimensional vectors for each text instance.

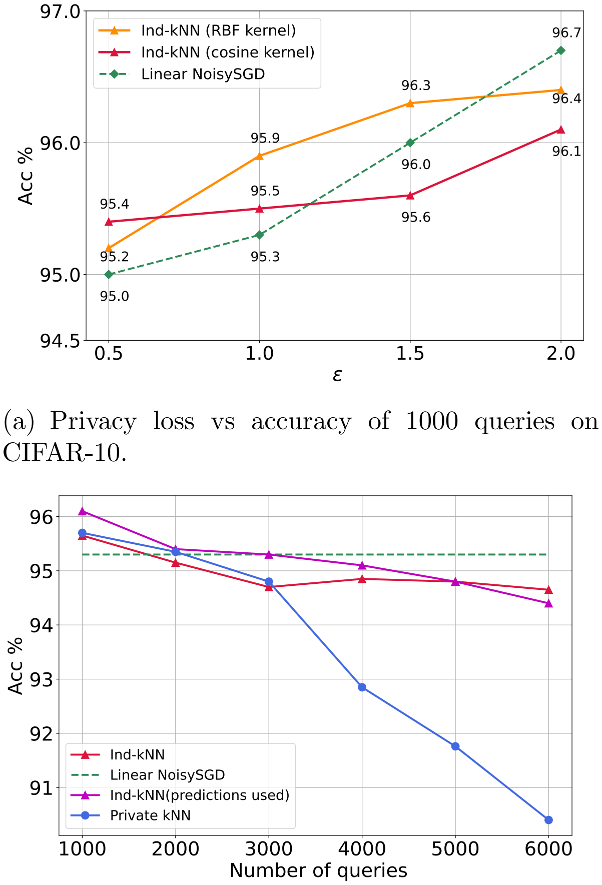
We consider the following two algorithms for comparisons:

Linear+NoisySGD [Tramèr and Boneh, 2021] is a private training benchmark that has been shown outperforming end-to-end privacy-preserving deep learning methods (including those pre-trained on public data, see De et al. [2022]) for a wide range of ϵ. We consider this algorithm as a reference point for private training to investigate how well Ind-KNN performs compared to private training while we gain those computational savings. We implement the algorithm by training a linear model with features extracted from the same extractor as Ind-KNN. We use the default batch size 256 and clip the gradient norm to 0.1. The model is trained for 10 epochs with a grid search over the learning rate and the noise level is determined by the target privacy budget.

Private-kNN [Zhu et al., 2020] is a private prediction baseline that we consider. For each query, the algorithm first samples a random subset from the private dataset, retrieves the k-nearest neighbors from the subset (based on the extracted features), and then releases the noisy label of kNN prediction using Report-Noisy-Max. We tune the sampling ratio and the number of neighbors on the validation set. The noise scale is calibrated based on the target privacy budget.

Hyper-parameters of Ind-KNN. We set the individual RDP budget B such that using RDP to DP conversion on $(\alpha, B\alpha)$-RDP satisfies the predefined privacy budget (ϵ, δ). Then, we set the noise scale σ_1 to be $\sqrt{\frac{T}{6B}}$ to use roughly half of the individual RDP budget B for each data point being selected at every query and tune the noise scale σ_2 on the validation set. We consider two kernel methods, the RBF kernel $\kappa(x, q) = e^{\frac{-||\phi(x)-\phi(q)||_2^2}{\nu^2}}$ and the cosine similarity $\kappa(x, q) = \cos(\phi(x), \phi(q))$. A linear scaling search is run on the minimum kernel weight threshold τ for each kernel method.

Experiment setting. For all experiments, we use a random seed to generate a validation set of size T. For example, we randomly sample 1000 examples from the CIFAR-10 testing dataset and tune the best hyper-parameters of all approaches on the validation set. We then report the median accuracy across 5 independent sampled query sets. All experiments are conducted on a server with an Intel i7-5930K CPU @ 3.50GHz and Nvidia TITAN Xp GPU.



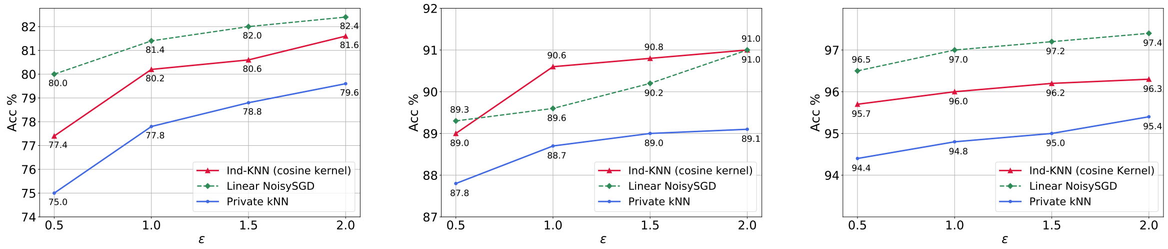
(a) Privacy loss vs accuracy of 1000 queries on CIFAR-10.

(b) Accuracy vs number of query on CIFAR-10 under $(1, 10^{-5})$-DP

Figure 5.1: Privacy-utility trade-offs on CIFAR-10. We plot the median accuracy across 5 independent runs.

5.4.1 Main Results

Privacy-accuracy trade-off on CIFAR-10. In the top figure of Figure 5.1, we plot the median accuracy evaluated on 1000 randomly chosen queries from the CIFAR-10 test set over a range of privacy budget ϵ. The hyper-parameters were fine-tuned for each algorithm at each value of ϵ. For Ind-kNN, we found that the best hyper-parameter τ (the minimum threshold) increases as the privacy budget grows. We note this because, with smaller value of ϵ, the added noise requires a larger margin among the selected neighbors' votes to determine the correct output. This larger margin, in turn, corresponds to a



(a) Accuracy of 500 queries on FMNIST.

(b) Accuracy of 800 queries on AG News.

(c) Accuracy of 800 queries on DBPedia.

Figure 5.2: Privacy-accuracy trade-offs on FMNIST, AG News and DBPedia. We consider $\delta = 10^{-5}$ for FMNIST and AG News and $\delta = 10^{-6}$ for DBPedia.

smaller threshold and more selected neighbors. For Ind-KNN with RBF kernel, we set the kernel bandwidth to $\nu = e^{1.5}$ and search for the optimal minimum threshold τ on the validation set. We find that different choices of kernel bandwidth in the RBF kernel produce similar accuracy results. As shown in Figure 5.1, Ind-kNN with RBF kernel performs slightly better than its cosine kernel and both kernel methods are comparable to Linear NoisySGD across various value of ϵ.

Accuracy vs number of queries on CIFAR-10. Given a fixed privacy budget, the accuracy of all private prediction methods typically degrades as the number of predictions increases, while the accuracy of private training methods remains unaffected. In the bottom figure of Figure 5.1, we study how quickly the accuracy of Ind-KNN drops as the number of queries increases. We present the median accuracy of answering T queries over five independent rounds. The accuracy of Private kNN drops rapidly with the increasing number of queries. This decline is expected, as Private kNN applies the standard Rényi composition theorem to analyze privacy loss, requiring the noise level to increase proportionally to the square root of T. In contrast, Ind-KNN uses individual privacy accountants, which only require selected neighbors to account for privacy loss, resulting in no significant accuracy drop as more queries are answered. Furthermore, exploiting released predictions allows Ind-KNN to answer an additional 1000 queries (from $T = 2000$ to $T = 3000$) without an accuracy drop. The figure also shows that if the number of queries is less than 2000, Ind-KNN can in fact outperform Linear NoisySGD, making it a practical alternative to private training methods when only a small number of predictions is needed.

Privacy-accuracy trade-off on Fashion MNIST, AG News and DBPedia. Next, we examine the privacy-accuracy trade-off on Fashion MNIST, AG News and DBPedia datasets. We use Ind-KNN with cosine kernel for all datasets. Figure 5.2(a) shows that Ind-KNN outperforms Private-kNN for all values of the privacy parameter ϵ on Fashion MNIST. On AG News, we compare the performance of Ind-KNN to that of Linear NoisySGD, and the results are presented in Figure 5.2(b). We evaluate $T = 800$ queries on AG News and find that the accuracy of Ind-KNN either surpasses or matches that of Linear NoisySGD for $\epsilon \geq 0.5$. We also observe similar improvements over Private-kNN on DBPedia.

Overall, Ind-KNN demonstrates its versatility by delivering competitive accuracy results on all three datasets, making it a promising solution for balancing differential privacy and accuracy.

5.4.2 Ablation Studies

We first perform an ablation study in Figure 5.3 to better understand how the periodical retraining affects the performance of private training method and our Ind-KNN in terms of computational and privacy cost on CIFAR-10.

Periodical retraining. In Figure **??**, we provide empirical measurements of the amortized computational cost associated with periodical retraining on CIFAR-10 of answering a stream of total $T = 10^5$ queries. We assume a retraining request is triggered every time the model has answered Q queries. To simplify the analysis, we assume each retraining is performed on the same dataset. For Linear NoisySGD, we retrain the model for 10 epochs and we calculate the per-query computational cost by dividing the total time spent on retraining and answering T queries by T. This provides an estimate of the average time required to answer a single query. For Ind-KNN with the cosine kernel, the average time of making predictions with is reported. Ind-KNN-Hash uses 30 hash tables with the width parameter $b = 8$. Our results demonstrate that the computational cost per query remains constant for Ind-KNN and Ind-KNN-Hash, as they do not require retraining the model, and the time required to add or delete individual data points is negligible. In contrast, for Linear NoisySGD, every retraining request incurs a substantial computational cost and the privacy loss grows $\propto \frac{1}{\sqrt{Q}}$ (proportional to the square root of total epochs). These findings highlight the advantage of Ind-KNN and Ind-KNN-Hash over Linear NoisySGD in terms of efficiency and resource utilization for machine unlearning and other scenarios with periodic retraining requests.

Figure 5.3(b) evaluates the accumulated privacy loss of answering a stream of $T = 2000$ queries on CIFAR-10. We tune hyper-parameters for both approaches such that the averaged accuracy of answering T queries is aligned to 96.0%. We consider two types of retraining scenarios: $Q = 100$ and $Q = 200$. Periodic retraining has a negligible privacy impact on Ind-KNN. Therefore, we only use one red curve to indicate the privacy loss of Ind-KNN under two scenarios. The individual privacy budget of Ind-KNN is predetermined, thus the standard privacy guarantee remained unchanged when making more predictions. The yellow curve plots the median of individual privacy loss over all private data points and reflects how much individual privacy loss deteriorates as the number of answered queries increases. We note the median individual privacy loss is $\epsilon = 1.2$ after answering 2000 queries, which suggests that only half of the privacy budget has been spent at an individual level. The privacy loss curve of Ind-KNN and two Linear NoisySGDs are met when there received six retraining requests. This suggests that if there are more than six retraining requests among the 2000 queries, the privacy loss of Ind-KNN would be better than that of Linear NoisySGD.

Table 5.2: Test Accuracy of $T = 1000$ queries on CIFAR-10 under different pre-trained models: vision transformer (ViT) [Dosovitskiy et al., 2020], SimCLRv2 model [Chen et al., 2020] and ResNet50 [He et al., 2016].

$\epsilon(\delta = 10^{-5})$	Method	ResNet50	SimCLRv2	ViT
$\epsilon = 0.5$	Linear NoisySGD	86.2%	89.7%	95.0%
	Private kNN	73.1%	76.0%	94.4%
	Ind-kNN	79.4%	82.4%	95.2%
$\epsilon = 2.0$	Linear NoisySGD	88.4%	90.2%	96.7%
	Private kNN	81.6%	84.7%	96.3%
	Ind-kNN	82.8%	86.3%	96.4%
$\epsilon = \text{inf}$	Linear NoisySGD	90.0%	90.7%	97.0%
	Private kNN	82.9%	85.1%	96.6%
	Ind-kNN	84.7%	89.2%	96.9%

Table 5.3: The Averaged time (in second) to answer each query on CIFAR-10 and AG News using Ind-KNN and its hashing variants.

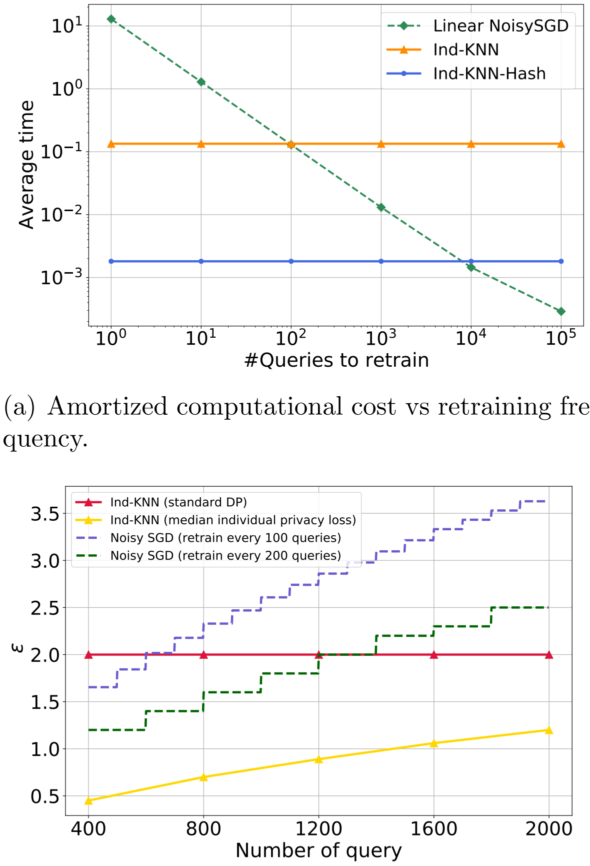
Dataset	Table=10	Table=20	Table=30	Ind-KNN
CIFAR-10	0.02	0.03	0.04	0.25
AG News	0.01	0.02	0.03	0.29

Ablation study on hashing. In Sec 5.3, we introduce hashing to improve the computational efficiency of Ind-KNN. We now investigate the trade-off between computational cost and utility of Ind-KNN-Hash on CIFAR-10 and AG News. We set the width parameter $b = 8$ for CIFAR-10 and $b = 9$ for AG News, and evaluate the performance of Ind-KNN-Hash with varying number of hash tables. As shown in Figure 5.4 and Table 5.3, the accuracy of hashing variants increases with more hash tables and more computational cost. We note that the computational cost roughly grows linearly with the number of the hash table. In particular, Ind-KNN with 30 hash tables matches the accuracy of the original Ind-KNN for a wide range of epsilon on CIFAR-10 but reduces the running time per query from 0.25 second to 0.03 second. Figure 5.2(b) shows similar observations for AG News.

Ablation study on the feature extractor. The quality of the feature extractor plays an crucial role in all three pre-trained feature-based methods. Remarkably, With the ViT feature extractor, even the Private kNN achieves an impressive accuracy of 96.3% at $\epsilon = 2.0$ on CIFAR-10, surpassing the previously reported best result of 95.4% [De et al., 2022] achieved using Wide-ResNets. Next, we present an ablation study focusing on three feature extractors and investigate the efficiency of each method on the CIFAR-10 task. Specifically, we consider three widely used vision models: vision transformer (ViT) [Dosovitskiy et al., 2020], the SimCLRv2 model [Chen et al., 2020] and ResNet 50 [He et al., 2016]. The SimCLRv2-based feature extractor has been considered by prior work Linear NoisySGD (Tramèr and Boneh [2021]), which trains a ResNet model on unlabeled ImageNet using SimCLRv2 model and provides a 4096-dim feature for each input image. For Resnet50, we consider the publicly-available ImageNet-pretrained Reset50 from Pytorch, which achieves a non-private accuracy at 90.0% for LinearSGD. As shown in Table 5.2, we find that Linear NoisySGD outperforms Private kNN and our Ind-kNN across ResNet50 and SimCLRv2. However, the performance gap decreases when applying a better feature extractor. This can be explained by the fact of their non-private performance. We also note that Private kNN is more fragile when ϵ is small, which could be due to its “loose” privacy analysis. Meanwhile, Ind-kNN handle the setting of small ϵ nicely, and can sometimes outperform Linear NoisySGD with a good feature extractor.

5.5 Conclusion

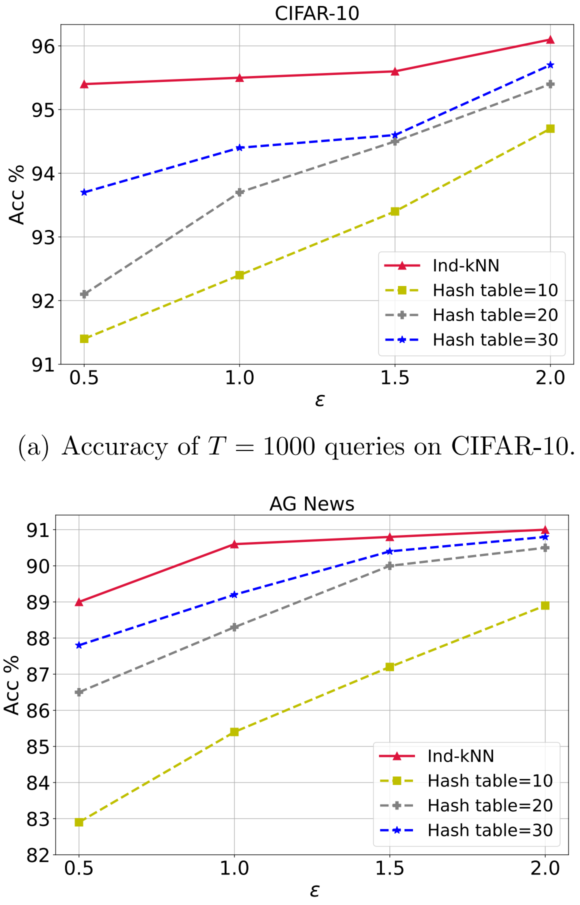
We propose a new algorithm, Individual Kernelized Nearest Neighbor (Ind-KNN), for private prediction in machine learning that is more flexible and updatable over dataset changes than private training. By modifying the KNN prediction and leveraging individualized privacy accountants, Ind-KNN allows a precise control of privacy at an individual level. Through extensive experimentation on four datasets, we demonstrate that Ind-KNN outperforms prior work Private kNN in terms of privacy and utility trade-offs. Furthermore, Ind-KNN exhibits superior computational efficiency and utility when dealing with frequent data updates, surpassing the private training method.



(a) Amortized computational cost vs retraining frequency.

(b) Privacy cost vs |queries| when accuracy is aligned to 96.0%.

Figure 5.3: (a): We estimate the amortized computational cost by averaging the time (in seconds) spent to answer each query under different retrain settings on CIFAR-10. The x-axis denotes the retraining frequency, i.e., retraining a model every receiving Q queries. (b): The accumulated privacy cost of answering a stream of $T = 2000$ queries when the final accuracy (over 2000 queries) is aligned to 96.0% on CIFAR-10. The red curve fixed the individual privacy budget at the beginning, resulting in a constant privacy loss. The yellow curve reports the median of individual privacy loss across all private data.



(a) Accuracy of $T = 1000$ queries on CIFAR-10.

(b) Accuracy of $T = 1000$ queries on AG News.

Figure 5.4: Ablation study on hashing under $\delta = 10^{-5}$.

Part III

Making classic DP mechanisms practical via data-adaptive analysis

Chapter 6

Generalized Propose-Test-Release (PTR)

6.1 Introduction

The guarantees of differential privacy (DP) [Dwork et al., 2006] are based on worst-case outcomes across all possible datasets. A common paradigm is therefore to add noise scaled by the *global sensitivity* of a query f, which measures the maximum change in f between any pair of neighboring datasets.

A given dataset X might have a *local sensitivity* $\Delta_{LS}(X)$ that is much smaller than the global sensitivity Δ_{GS}, in which case we can hope to add a smaller amount of noise (calibrated to the local rather than global sensitivity) while achieving the same privacy guarantee. This must not be undertaken naïvely; the local sensitivity is a dataset-dependent function and so calibrating noise to the local sensitivity could leak information about the dataset [Nissim et al., 2007].

The "Propose-Test-Release" (PTR) framework [Dwork and Lei, 2009] resolves this issue by introducing a test to privately check whether a proposed bound on the local sensitivity is valid. Only if the test "passes" is the output released with noise calibrated to the proposed bound on the local sensitivity.

PTR is a powerful tool for designing data-adaptive DP algorithms, but it has several limitations. First, it applies only to noise-adding mechanisms which calibrate noise according to the sensitivity of a query. Second, the test in "Propose-Test-Release" is computationally expensive for all but a few simple queries such as privately releasing the median or mode. Third, while some existing works [Decarolis et al., 2020, Kasiviswanathan et al., 2013, Liu et al., 2021] follow the adaptive approach of privately testing properties of an input dataset for "niceness"[1], there has not been a systematic recipe for *discovering* which properties should be tested.

In this paper, we propose a generalization of PTR which addresses these limitations. The centerpiece of our framework is a differentially private test on the *data-dependent privacy loss*. This test does not directly consider the local sensitivity of a query and is therefore not limited to additive noise mechanisms. Moreover, in many cases the test can be efficiently implemented by privately releasing a high-probability upper bound, thus avoiding the need to search an exponentially large space of datasets. Furthermore, the derivation of the test itself often spells out exactly what properties of the input dataset need to be checked, which streamlines the design of data-adaptive DP algorithms.

Our contributions are summarized as follows:

1. We propose a generalization of PTR which can handle algorithms beyond noise-adding mechanisms. Generalized PTR allows us to plug in *any* data-dependent DP analysis to construct a high-probability DP test that adapts to favorable properties of the input dataset, without painstakingly designing each test from scratch.

2. We show that many existing examples of PTR and PTR-like methods can be unified

[1] We refer to these as PTR-like algorithms.

under the generalized PTR framework, sometimes resulting in a tighter analysis (see an example of report-noisy-max in Section 6.5).

3. We demonstrate that one can publish a DP model through privately upper-bounding a one-dimensional statistic — no matter how complex the output space of the mechanism is. We apply this result to solve an open problem from PATE [Papernot et al., 2017, 2018].

4. Our results broaden the applicability of private hyperparameter tuning [Liu and Talwar, 2019, Papernot and Steinke, 2021] in enabling joint selection of DP-specific parameters (e.g., noise level) and native parameters of the algorithm (e.g., regularization).

6.2 Preliminaries

Datasets $X, X' \in \mathcal{X}$ are neighbors if they differ by no more than one datapoint; we say $X \simeq X'$ if $d(X, X') \leq 1$.

We measure the distance $d(\cdot)$ between same-sized datasets $X = \{x_i\}_{i=1}^n$ and $\tilde{X} = \{\tilde{x}_i\}_{i=1}^n$ as the number of coordinates that differ between them:

$$d(X, \tilde{X}) = \#\{i \in [n] : x_i \neq \tilde{x}_i\}.$$

We use $||\cdot||$ to denote the radius of the smallest Euclidean ball that contains the input set, e.g. $||\mathcal{X}|| = \sup_{x \in \mathcal{X}} ||x||$.

For mechanisms with continuous output space, the probability density of $\mathcal{M}(X)$ at y is denoted $\Pr[\mathcal{M}(X) = y]$.

[2] This is probably folklore. We could not find the particular approach with AboveThreshold presented in the literature — the original PTR work by Dwork and Lei [2009] uses composition, thus depends on poly(M), while using AboveThreshold (or our approach with general DP selection) incurs only log(M).

	PTR	Generalized PTR
Private Test	Test $\Delta_{\text{LS}} \leq \beta$ for a proposed bound β, then add noise $\propto \beta$ if the test passes [Vadhan, 2017, Sec 3.2].	Test $\epsilon_\phi \leq \epsilon$ for a proposed parameter ϕ, then run $\mathcal{M}_\phi$ if the test passes (Alg 7)
Private pointwise bounds	no analogous algorithm	Release $\bar{\epsilon}$ s.t. $\epsilon_\phi \leq \bar{\epsilon}$ for a *fixed* ϕ w.h.p. for general randomized mechanism $\mathcal{M}_\phi$, then run $\mathcal{M}_\phi$ if $\bar{\epsilon} \leq \epsilon$ (Alg 7).
Private uniform bounds	Release $\bar{\Delta}$ s.t. $\Delta_{\text{LS}} \leq \bar{\Delta}$ w.h.p for a noise-adding mechanism with noise $\propto \bar{\Delta}$ [Vadhan, 2017, Sec 3.4]. (Choose appropriate noise level σ, no $\perp$.)	Release $\bar{\epsilon}_\phi$ s.t. $\epsilon_\phi \leq \bar{\epsilon}_\phi$ for *all* ϕ w.h.p. for general randomized mechanism $\mathcal{M}_\phi$ (Choose appropriate ϕ, no $\perp$, as in Alg **??**)
Stability-based	Test $\Delta_{\text{LS}} = 0$ before releasing stable numerical value deterministically [Vadhan, 2017, Sec 3.3].	Test $\epsilon_\phi = 0$ before releasing stable general output deterministically (special case of Alg 7).
What to propose?	Select $\beta \in \{\beta_1, ..., \beta_M\}$ s.t. $\Delta_{\text{LS}} \leq \beta$ passes the test (using e.g. AboveThreshold)[2]	Select $\phi \in \{\phi_1, ..., \phi_M\}$, s.t. ϵ_ϕ passes the test (using private selection as in Alg 8).

Table 6.1: A summary of our generalization to the standard variants of PTR. The vanilla PTR, often implemented using a distance test was proposed originally in Dwork and Lei [2009]. The stability-based argument was originally proposed by Thakurta and Smith [2013]. We are citing the book of Vadhan [2017] for a clean treatment to these PTR-like mechanisms. The corresponding generalized version are from this paper.

Definition 6.2.1 (Sensitivity)**.** The global ℓ_*-sensitivity of a function f is defined as

$$\Delta_{GS} = \max_{X,X':X\simeq X'} ||f(X) - f(X')||_*$$

and its local sensitivity at dataset X is

$$\Delta_{LS}(X) = \max_{X\simeq X'} ||f(X) - f(X')||_*.$$

6.2.1 Propose-Test-Release

Calibrating the noise level to the local sensitivity $\Delta_{LS}(X)$ of a function would allow us to add less noise and therefore achieve higher utility for releasing private queries. However, the local sensitivity is a data-dependent function and naïvely calibrating the noise level to $\Delta_{LS}(X)$ will not satisfy DP.

PTR resolves this issue in a three-step procedure: **propose** a bound on the local sensitivity, privately **test** that the bound is valid (with high probability), and if so calibrate noise according to the bound and **release** the output.

PTR privately computes the distance $\mathcal{D}_\beta(X)$ between the input dataset X and the nearest dataset X'' whose local sensitivity exceeds the proposed bound β:

$$\mathcal{D}_\beta(X) = \min_{X''}\{d(X, X'') : \Delta_{LS}(X'') > \beta\}.$$

Algorithm 6 Propose-Test-Release [Dwork and Lei, 2009]

```
1: **Input**: Dataset $X$; privacy parameters $\epsilon, \delta$; proposed bound $\beta$; query function $f : \mathcal{X} \to \mathbb{R}$.
2: **if** $\mathcal{D}_\beta(X) + \text{Lap}\left(\frac{1}{\epsilon}\right) \leq \frac{\log(1/\delta)}{\epsilon}$ **then** output $\perp$,
3: **else** release $f(X) + \text{Lap}\left(\frac{\beta}{\epsilon}\right)$.
```

Theorem 6.2.2 (PTR [Dwork and Lei, 2009])**.** *Algorithm 6 satisfies* $(2\epsilon, \delta)$*-DP.*

Rather than proposing an arbitrary bound β on $\Delta_{LS}(X)$, one can also privately release an upper bound of the local sensitivity and calibrate noise according to this upper bound. This was used for node DP in graph statistics [Kasiviswanathan et al., 2013], and for fitting topic models using spectral methods [Decarolis et al., 2020].

6.3 Related Work

Data-dependent DP algorithms. Privately calibrating noise to the local sensitivity is a well-studied problem. One approach is to add noise calibrated to the smooth sensitivity [Nissim et al., 2007], an upper bound on the local sensitivity which changes slowly between neighboring datasets. An alternative to this — and the focus of our work — is Propose-Test-Release (PTR) [Dwork and Lei, 2009], which works by calculating the distance $\mathcal{D}_\beta(X)$ to the nearest dataset to X whose local sensitivity violates a proposed bound β. The PTR algorithm then adds noise to $\mathcal{D}_\beta(X)$ before testing whether this privately computed distance is large enough to permit releasing the output with noise calibrated to β.

PTR spin-offs abound. Notable examples include stability-based methods [Thakurta and Smith, 2013] (stable local sensitivity of 0 near the input data) and privately releasing upper bounds of local sensitivity [Kasiviswanathan et al., 2013, Liu et al., 2021, Decarolis et al., 2020]. We refer readers to Chapter 3 of Vadhan [2017] for a concise summary of these classic results. More recently, Wang et al. [2022] have provided Rényi DP bounds [Mironov, 2017] for PTR and demonstrated its applications to robust DP-SGD. Our work (Section 6.4.6) also considers applications of PTR in data-adaptive private deep learning: Instead of testing the local sensitivity of each gradient step as in Wang et al. [2022], our PTR-based PATE algorithm tests the data-dependent privacy loss as a whole.

Liu et al. [2021] proposed the High-dimensional Propose-Test-Release (HPTR) framework. HPTR provides a systematic way of solving DP statistical estimation problems by using the exponential mechanism (EM) with carefully constructed scores based on certain one-dimensional robust statistics, which have stable local sensitivity bounds. HPTR focuses on designing data-adaptive DP mechanisms from scratch; our method, in contrast, converts existing randomized algorithms (including EM and even some that do not satisfy DP) into those with formal DP guarantees. Interestingly, our proposed method also depends on a one-dimensional statistic of direct interest: the data-dependent privacy loss.

Data-dependent DP losses. The flip side of data-dependent DP algorithms is the study of data-dependent DP losses [Papernot et al., 2018, Soria-Comas et al., 2017, Wang, 2018a], which fix the randomized algorithm but parameterize the resulting privacy loss by the specific input dataset. For example: In the simple mechanism that adds Laplace noise with parameter b, data-dependent DP losses are $\epsilon(X) = \Delta_{LS}(X)/b$. The data-dependent DP losses $\epsilon(X)$ are often much smaller than the DP loss ϵ, but they themselves depend on the data and thus may reveal sensitive information; algorithms satisfying a data-dependent privacy guarantee are not formally DP with guarantees any smaller than that of the worst-case. Existing work has considered privately publishing these data-dependent privacy losses [Papernot et al., 2018, **?**, Redberg and Wang, 2021], but notice that privately publishing these losses does not improve the DP parameter of the given algorithm. Part of our contribution is to resolve this conundrum by showing that a simple post-processing step of the privately released upper bound of $\epsilon(X)$ gives a formal DP algorithm.

Private hyperparameter tuning. Our work has a nice connection with private hyperparameter tuning. Prior work [Liu and Talwar, 2019, Papernot and Steinke, 2021] requires each candidate configuration to be released with the same DP (or Rényi DP) parameter set. Another hidden assumption is that the parameters must not be privacy-correlated (i.e., parameter choice will not change the privacy guarantee). Otherwise we need to use the largest DP bound across all candidates. For example, Liu and Talwar [2019] show that if each mechanism (instantiated with one group of hyperparameters) is $(\epsilon, 0)$-DP, then running a random number of mechanisms and reporting the best option satisfies $(3\epsilon, 0)$-DP. Our work directly generalizes the above results by (1) considering a wide range of hyperparameters, either privacy-correlated or not; and (2) requiring only that individual candidates have a *testable* data-dependent DP.

6.4 Main results: Generalized PTR

This section introduces the generalized PTR framework. We first formalize the notion of *data-dependent* differential privacy that conditions on an input dataset X.

Definition 6.4.1 (Data-dependent privacy)**.** Suppose we have $\delta > 0$ and a function $\epsilon : \mathcal{X} \to \mathbb{R}^+$. We say that mechanism $\mathcal{M}$ satisfies $(\epsilon(X), \delta)$ data-dependent DP[3] for dataset X if for all possible output sets S and neighboring datasets X',

$$\Pr\big[\mathcal{M}(X) \in S\big] \leq e^{\epsilon(X)} \Pr\big[\mathcal{M}(X') \in S\big] + \delta,$$
$$\Pr\big[\mathcal{M}(X') \in S\big] \leq e^{\epsilon(X)} \Pr\big[\mathcal{M}(X) \in S\big] + \delta.$$

In generalized PTR, we propose a value (or set of values) ϕ with which to parameterize mechanism $\mathcal{M}_\phi$. For instance, in Example 6.4.4 we might propose $\phi = (\gamma, \lambda)$ as a parameter set that includes the noise scale and regularization strength. For a given δ, we then say that mechanism $\mathcal{M}_\phi$ satisfies $\epsilon_\phi(X)$ data-dependent DP for dataset X.

The following example illustrates how to derive the data-dependent DP for a familiar friend – the Laplace mechanism.

Example 6.4.2. (*Data-dependent DP of Laplace Mechanism.*) Given a function $f : \mathcal{X} \to \mathbb{R}$, we will define

$$\mathcal{M}_\phi(X) = f(X) + \mathrm{Lap}\left(\phi\right).$$

[3] We will sometimes write that $\mathcal{M}(X)$ satisfies $\epsilon(X)$ data-dependent DP w.r.t. δ.

We then have

$$\log \frac{Pr[\mathcal{M}_\phi(X) = y]}{Pr[\mathcal{M}_\phi(X') = y]} \leq \frac{|f(X) - f(X')|}{\phi}.$$

Maximizing over all possible outputs y yields an equality between the two expressions above. Using Definition 6.4.1,

$$\epsilon_\phi(X) = \max_{X':X\simeq X'} \frac{|f(X) - f(X')|}{\phi} = \frac{\Delta_{LS}(X)}{\phi}.$$

Maximizing $\epsilon_\phi(X)$ over X recovers the standard DP guarantee of running $\mathcal{M}$ with parameter ϕ.

Algorithm 7 distills the generalized PTR framework into a simple procedure: we run mechanism $\mathcal{M}$ with proposed parameter ϕ only if the test $\mathcal{T}$ "passes".

Let's suppose that our privacy budget for mechanism $\mathcal{M}_\phi$ is (ϵ, δ); that our test $\mathcal{T}$ satisfies $(\hat{\epsilon}, \hat{\delta})$-DP; and that $\mathcal{T}$ has a "false positive" rate δ', meaning $\mathcal{T}$ passes an insufficient proposal ϕ (where $\mathcal{M}_\phi$ exceeds its privacy budget) with probability at most δ'. Theorem 6.4.3 states the privacy guarantee of generalized PTR under these assumptions.

Algorithm 7 Generalized Propose-Test-Release

```
Input: Dataset $X$; mechanism $\mathcal{M}_\phi : \mathcal{X} \to \mathcal{R}$ and its privacy budget $\epsilon, \delta$; $(\hat{\epsilon}, \hat{\delta})$-DP test $\mathcal{T}$; false positive rate $\leq \delta'$; data-dependent DP function $\epsilon_\phi(\cdot)$ w.r.t. $\delta$.
if not $\mathcal{T}(X)$ then output $\perp$,
else release $\theta = \mathcal{M}_\phi(X)$.
```

Theorem 6.4.3 (Privacy guarantee of generalized PTR)**.** *Consider a proposal ϕ and a data-dependent DP function $\epsilon_\phi(X)$ w.r.t. δ. Suppose that we have an $(\hat{\epsilon}, \hat{\delta})$-DP test*

$\mathcal{T} : \mathcal{X} \to \{0,1\}$ *such that when* $\epsilon_\phi(X) > \epsilon$,

$$\mathcal{T}(X) = \begin{cases} 0 & \textit{with probability } 1-\delta', \\ 1 & \textit{with probability } \delta'. \end{cases}$$

Then Algorithm 7 satisfies $(\epsilon + \hat{\epsilon}, \delta + \hat{\delta} + \delta')$*-DP.*

Proof: [Proof sketch] We can split the possible input datasets X into two main cases based on the data-dependent DP for a given δ: $\epsilon_\phi(X) > \epsilon$ and $\epsilon_\phi(X) \le \epsilon$. At a high level, we can analyze both cases using the composition property of DP (that ϵ's and δ's "add up") and then combine them by taking an upper bound of the maximum value of the ϵ's and δ's between the two cases.

By the "false positive" assumption on the test $\mathcal{T}$, the first case can be viewed as a composition of an $(\hat{\epsilon}, \hat{\delta})$-DP mechanism and a $(0, \delta')$-DP mechanism. The second case, when the data-dependent DP is at most ϵ, is a composition of an $(\hat{\epsilon}, \hat{\delta})$-DP mechanism and an (ϵ, δ)-DP mechanism.

■

Generalized PTR is a *strict* generalization of Propose-Test-Release. For some function f, define $\mathcal{M}_\phi$ and $\mathcal{T}$ as follows:

$$\mathcal{M}_\phi(X) = f(X) + \mathrm{Lap}(\phi);$$
$$\mathcal{T}(X) = \begin{cases} 0 & \text{if } \mathcal{D}_\beta(X) + \mathrm{Lap}\left(\frac{1}{\epsilon}\right) > \frac{\log(1/\delta)}{\epsilon}, \\ 1 & \text{otherwise.} \end{cases}$$

Notice that our choice of parameterization is now $\phi = \frac{\beta}{\epsilon}$, where ϕ is the scale of the Laplace noise. In other words, we know from Example 6.4.2 that $\epsilon_\phi(X) > \epsilon$ exactly when $\Delta_{LS}(X) > \beta$.

For noise-adding mechanisms such as the Laplace mechanism, the sensitivity is proportional to the privacy loss in both the global and local sense: $\Delta_{GS} \propto \epsilon$ and $\Delta_{LS}(X) \propto \epsilon(X)$. Therefore for these mechanisms the only difference between privately testing the local sensitivity (Algorithm 6) and privately testing the data-dependent DP (Theorem 6.4.3) is a change of parameterization.

6.4.1 Limitations of local sensitivity

Why do we want to generalize PTR beyond noise-adding mechanisms? Compared to classic PTR, the generalized PTR framework allows us to be more flexible in both the type of test conducted and also the type of mechanism whose output we wish to release. For many mechanisms, the local sensitivity either does not exist or is only defined for specific data-dependent quantities (e.g., the sensitivity of the score function in the exponential mechanism) rather than the mechanism's output.

The following example illustrates this issue.

Example 6.4.4 (Private posterior sampling). Let $\mathcal{M} : \mathcal{X} \times \mathcal{Y} \to \Theta$ be a private posterior sampling mechanism [Minami et al., 2016, Wang et al., 2015, Gopi et al., 2022] for approximately minimizing $F_X(\theta)$.

$\mathcal{M}$ samples $\theta \sim P(\theta) \propto e^{-\gamma(F_X(\theta)+\lambda/2||\theta||_2^2)}$ with parameters γ, λ. Note that γ, λ cannot be appropriately chosen for this mechanism to satisfy DP without calculating the sensitivity of $\arg\min F_X(\theta)$, which in many cases (e.g., logistic regression) lacks a closed-form solution. In fact, the global and local sensitivity of the minimizer is unbounded even in linear regression problems, i.e when $F_X(\theta) = \frac{1}{2}||y - X\theta||_2^2$.

Output perturbation algorithms do work for the above problem when we regularize, but they are known to be suboptimal in theory and in practice [Chaudhuri et al., 2011]. In Section 6.4.5 we demonstrate how to apply generalized PTR to achieve a data-adaptive posterior sampling mechanism.

Even in the cases of noise-adding mechanisms where PTR seems to be applicable, it does not lead to a tight privacy guarantee. Specifically, by an example of privacy amplification by post-processing (Example 6.5.1 in the appendix), we demonstrate that the local sensitivity does not capture all sufficient statistics for data-dependent privacy analysis and thus is loose.

6.4.2 Which ϕ to propose

A limitation of generalized PTR (inherited from its predecessor) is that one needs to "propose" a good guess of parameter ϕ. Take the example of ϕ being the noise level in a noise-adding mechanism. Choosing too small a ϕ will result in a useless output $\perp$, while choosing too large a ϕ will add more noise than necessary. Finding this 'Goldilocks' ϕ might require trying out many different possibilities – each of which will consume privacy budget. This section introduces a method to jointly tune privacy parameters (e.g., noise scale) along with parameters related only to the utility of an algorithm (e.g., learning rate or batch size in stochastic gradient descent) — while avoiding the $\perp$ output.

Algorithm 8 takes a list of parameters as input, runs generalized PTR with each of the parameters, and returns the output with the best utility. We show that the privacy guarantee with respect to ϵ is independent of the number of ϕ that we try.

Formally, let $\phi_1, ..., \phi_k$ be a set of hyperparameters and $\tilde{\theta}_i \in \{\perp, \mathrm{Range}(\mathcal{M})\}$ the output of running generalized PTR with ϕ_i on dataset X. Let X_{val} be a public validation set and $q(\tilde{\theta}_i)$ be the score of evaluating $\tilde{\theta}_i$ with X_{val} (e.g., validation accuracy). The goal is to select a pair $(\tilde{\theta}_i,\, \phi_i)$ such that DP model $\tilde{\theta}_i$ maximizes the validation score.

The generalized PTR framework with privacy calibration is described in Algorithm 8; its privacy guarantee is an application of Liu and Talwar [2019].

Algorithm 8 PTR with hyperparameter selection

```
1: Input: Privacy budget per PTR algorithm $(\epsilon^*, \delta^*)$, cut-off $T$, parameters $\phi_{1:k}$, flipping probability $\tau$ and validation score function $q(\cdot)$.
2: Initialize the set $S = \varnothing$.
3: Draw $G$ from a geometric distribution $\mathcal{D}_\tau$ and let $\hat{T} = \min(T, G)$.
4: for i = 1 ,..., $\hat{T}$ do
5:     pick a random $\phi_i$ from $\phi_{1:k}$.
6:     evaluate $\phi_i$: $(\tilde{\theta}_i, q(\tilde{\theta}_i)) \leftarrow$ Algorithm 7$(\phi_i, (\epsilon^*, \delta^*))$.
7:     $S \leftarrow S \cup \{\tilde{\theta}_i, q(\tilde{\theta}_i)\}$.
8: end for
9: Output the highest scored candidate from $S$.
```

Theorem 6.4.5 (Theorem 3.4 [Liu and Talwar, 2019])**.** *Fix any $\tau \in [0,1], \delta_2 > 0$ and let $T = \frac{1}{\tau}\log\frac{1}{\delta_2}$. If each oracle access to Algorithm 7 is (ϵ^*, δ^*)-DP, then Algorithm 8 is $(3\epsilon^* + 3\sqrt{2\delta^*}, \sqrt{2\delta^*}T + \delta_2)$-DP.*

The theorem implies that one can try a random number of ϕ while paying a constant ϵ. In practice, we can roughly set $\tau = \frac{1}{10k}$ so that the algorithm is likely to test all k parameters. We emphasize that the privacy and the utility guarantee is not our contribution. But the idea of applying generalized PTR to enforce a uniform DP guarantee over all choices of parameters with a data-dependent analysis is new.

6.4.3 Construction of the DP test

Classic PTR uses the Laplace mechanism to construct a differentially private upper bound of $\mathcal{D}_\beta(X)$, the distance from input dataset X to the closest dataset whose local sensitivity exceeds the proposed bound β. The tail bound of the Laplace distribution then ensures that if $\mathcal{D}_\beta(X) = 0$ (that is, if $\Delta_{LS}(X) > \beta$), then the output will be released with only a small probability δ.

The following theorem shows that we could instead use a differentially private upper bound of the data-dependent DP $\epsilon_\phi(X)$ in order to test whether to run the mechanism

$\mathcal{M}_\phi$.

Theorem 6.4.6 (Generalized PTR with private upper bound)**.** *Suppose we have a differentially private upper bound of $\epsilon_\phi(X)$ w.r.t. δ such that with probability at least $1-\delta'$, $\epsilon^P_\phi(X) > \epsilon_\phi(X)$. Further suppose we have an $(\hat{\epsilon}, \hat{\delta})$-DP test $\mathcal{T}$ such that*

$$T(X) = \begin{cases} 1 & \textit{if } \epsilon^P_\phi(X) < \epsilon, \\ 0 & \textit{otherwise.} \end{cases}$$

Then Algorithm 7 is $(\epsilon + \hat{\epsilon}, \delta + \hat{\delta} + \delta')$-DP.

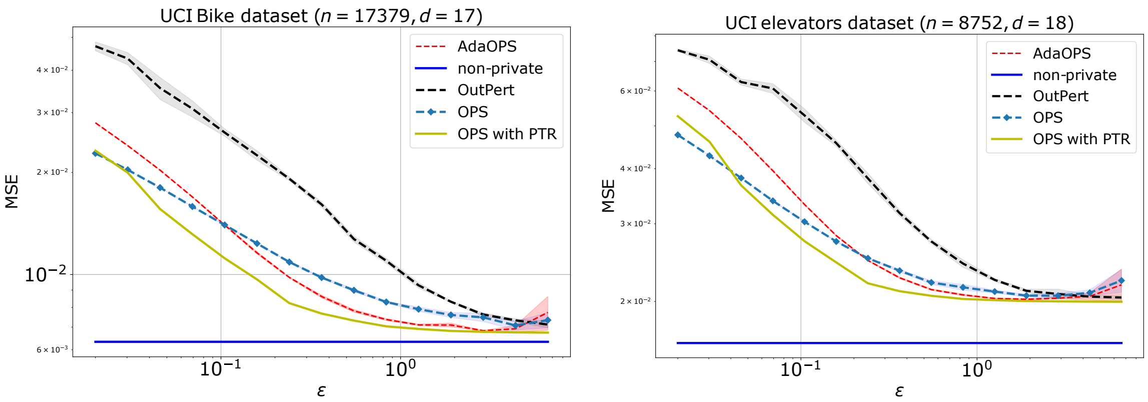
In Section 6.4.6, we demonstrate how to upper-bound the data-dependent DP through a modification of the smooth sensitivity framework applied on $\epsilon_\phi(X)$. In Section 6.4.5 we provide a direct application of Theorem 6.4.6 with private linear regression by making use of the per-instance DP technique [Wang, 2018a].

The applications in Section 6.4.4 are illustrative of two distinct approaches to constructing the DP test for generalized PTR:

1. Private sufficient statistics release (used in the private linear regression example of Section 6.4.5) specifies the data-dependent DP as a function of the dataset and privately releases each data-dependent component.

2. The second approach (used in the PATE example of Section 6.4.6) uses the smooth sensitivity framework to privately release the data-dependent DP as a whole, and then construct a high-confidence test using the Gaussian mechanism.

These two flavors cover most of the scenarios arising in data-adaptive analysis. For example, in the appendix we demonstrate the merits of generalized PTR in handling data-adaptive private generalized linear models (GLMs) using private sufficient statistics



(a) Bike dataset (b) Elevators dataset

Figure 6.1: Differentially private linear regression algorithms on UCI datasets. y-axis reports the MSE error with confidence intervals. ϵ is evaluated with $\delta = 1e^{-6}$.

release. Moreover, sufficient statistics release together with our private hyperparameter tuning (Algorithm 8) can be used to construct data-adaptive extensions of DP-PCA and Sparse-DP-ERM (see details in the future work section).

6.4.4 Applications

In this section, we put into action our approaches to construct the DP test and provide applications in private linear regression and PATE.

6.4.5 Private Linear Regression

Theorem 6.4.7 ([Wang, 2018a])**.** *For input data $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, define the following:*

- $\lambda_{\min}(X)$ *denotes the smallest eigenvalue of* $X^T X$*;*
- $||\theta^*_\lambda||$ *is the magnitude of the solution* $\theta^*_\lambda = (X^T X + \lambda I)^{-1} X^T Y$*;*
- *and* $L(X, Y) := ||\mathcal{X}||\big(||\mathcal{X}||\,||\theta^*_\lambda|| + ||\mathcal{Y}||\big)$ *is the local Lipschitz constant, denoted* L *in brief.*

For brevity, denote $\lambda^* = \lambda + \lambda_{\min}(X)$*. The algorithm used in Example 6.4.4 with proposal* $\phi = (\lambda, \gamma)$ *obeys* $(\epsilon_\phi(Z), \delta)$ *data-dependent DP for each dataset* $Z = (X, Y)$ *with* $\epsilon_\phi(Z)$ *equal to*

$$\sqrt{\frac{\gamma L^2 \log(2/\delta)}{\lambda^*}} + \frac{\gamma L^2}{2(\lambda^* + ||\mathcal{X}||^2)} + \frac{1 + \log(2/\delta)||\mathcal{X}||^2}{2\lambda^*}.$$

Notice that $\epsilon_\phi(Z)$ is a function of the data-dependent quantities $\lambda_{\min}(X)$ and L (which is itself a function of $||\theta^*_\lambda||$). Could we privately release $\epsilon_\phi(Z)$ and tune the privacy parameters $\phi = (\lambda, \gamma)$ based on the sanitized data-dependent DP? Unfortunately in this case, $||\theta^*_\lambda||$ is a complicated function of λ and it is not clear how to choose an optimal λ.

The calibration of γ, however, is fairly straightforward from the expression for $\epsilon_\phi(Z)$ given in Theorem 6.4.7. We can apply the generalized PTR framework to the private posterior sampling problem described in Example 6.4.4 by proposing $\phi = \lambda$ as the regularization parameter; releasing a high-probability upper bound $\epsilon^P_\lambda(Z)$ of the data-dependent DP, as a function of γ; and tuning the noise scale γ to achieve the desired utility under the constraint $\epsilon^P_\lambda(Z) \leq \epsilon$.

Example 6.4.8 (OPS for linear regression with PTR). Consider the posterior sampling mechanism described in Example 6.4.4 and the expression $\epsilon_\phi(Z)$ given in Theorem 6.4.7. Suppose we have a quality score $q(\cdot)$ that measures the utility of the input parameter, e.g. $q(\gamma) = \gamma$ for the inverse noise scale. We can apply generalized PTR as follows.

- Given a proposed value $\phi = \lambda$, privately release $\lambda_{\min}(X)$ and L with combined privacy budget $(\hat{\epsilon}, \hat{\delta})$ in order to obtain $\epsilon^P_\lambda(Z)$ such that with probability $1 - \delta'$, $\epsilon^P_\lambda(Z) \leq \epsilon_\lambda(Z)$.
- Calibrate $\gamma^* = \sup_{q(\gamma)}\{\gamma \mid \epsilon^P_\lambda(Z) \leq \epsilon\}$.
- Output $\theta \sim e^{-\frac{\gamma^*}{2}\left(||Y - X\theta||_2^2 + \lambda||\theta||_2^2\right)}$ if γ^* exists; else output $\perp$.

In the full paper, we provide full details of the above algorithm and show that it satisfies $(\epsilon + \hat{\epsilon}, \delta + \hat{\delta} + \delta')$-DP.

[Dwork et al., 2014c] provides a PTR style privacy-preserving principle component analysis (PCA). The key observation of [Dwork et al., 2014c] is that the local sensitivity is quite "small" if there is a large eigengap between the k-th and the $k+1$-th eigenvalues. Therefore, their approach (Algorithm 2) chooses to privately release a lower bound of the k-th eigengap (k is fixed as an input) and use that to construct a high-confidence upper bound of the local sensitivity.

For noise-adding mechanisms, the local sensitivity is proportional to the data-dependent loss and generalized PTR is applicable. We can formulate the data-dependent DP of DP-PCA as follows:

Theorem 6.4.9. *For a given matrix $A \in \mathcal{R}^{m \times n}$, assume each row of A has a bounded ℓ_2 norm being 1. Let V_k denotes the top k eigenvectors of A^TA and d_k denotes the gap between the k-th and the $k+1$-th eigenvalue. Then releasing $V_kV_k^T + E$, where $E \in \mathcal{R}^{n \times n}$ is a symmetric matrix with the upper triangle is i.i.d samples from $\mathcal{N}(0, \sigma^2)$ satisfies $(\epsilon(A), \delta)$ data-dependent DP and $\epsilon(A) = \frac{2\sqrt{\log(1.25/\delta)}}{\sigma(d_k - 2)}$.*

The proof is based on the local sensitivity result from [Dwork et al., 2014c] and the noise calibration of Gaussian mechanism.

We can combine Theorem 6.5.3 with our Algorithm 8 to instantiate the generalized PTR framework. The improvement over Dwork et al. [2014c] will be to allow joint tuning of the parameter k and the noise variance (added to the spectral gap d_k).

The main idea of the above algorithm boils down to privately releasing all data-dependent quantities in data-dependent DP, constructing high-probability confidence intervals of these quantities, and then deciding whether to run the mechanism $\mathcal{M}$ with the proposed parameters. In Example 6.4.4, we need only propose λ as we can tune γ

directly based on $\epsilon^P_\lambda(Z)$.

Remark 6.4.10. Tuning λ is even more troublesome for generalized linear models (GLMs) beyond linear regression. The data-dependent DP there involves a local strong-convexity parameter that is a complex function of the regularizer λ and for which we only have zeroth-order access. In the full paper, we demonstrate how to apply generalized PTR to provide a generic solution to a family of private GLMs where the link function satisfies a self-concordance assumption.

We next apply Algorithm 8 for Example 6.4.8 with UCI regression datasets. Standard z-scoring is applied and each data point is normalized with a Euclidean norm of 1. We consider $(60\%, 10\%, 30\%)$ splits for the train, validation and test sets.

Baselines

- Output Perturbation (Outpert) [Chaudhuri et al., 2011]: $\theta = (X^TX + \lambda I)^{-1}X^T\mathbf{y}$. Release $\hat{\theta} = \theta + \mathbf{b}$ with an appropriate λ, where $\mathbf{b}$ is a Gaussian random vector.
- Posterior sampling (OPS). Sample $\hat{\theta} \sim P(\theta) \propto e^{-\gamma(F(\theta)+0.5\lambda||\theta||^2)}$ with parameters γ, λ.
- Adaptive posterior sampling (AdaOPS) [Wang, 2018b]. Run OPS with (λ, γ) chosen adaptively according to the dataset.

Outpert and OPS serve as two non-adaptive baselines. In particular, we consider OPS-Balanced [Wang, 2018b], which chooses λ to minimize a data-independent upper bound of empirical risk and dominates other OPS variants. AdaOPS is one state-of-the-art algorithm for adaptive private regression, which automatically chooses λ by minimizing an upper bound of the data-dependent empirical risk.

We implement OPS-PTR as follows: propose a list of λ through grid search (we choose $k = 30$ and λ ranges from $[2.5, 2.5^{10}]$ on a logarithmic scale); instantiate Algorithm 8 with

$\tau = 0.05/k$, $T = \frac{1}{\tau}\log(1/\delta_2)$ and $\delta_2 = 1/2\delta$; calibrate the per-PTR privacy budget (ϵ^*, δ^*) according to Theorem 6.4.5; set $\epsilon = \hat{\epsilon} = 0.5\epsilon^*$ and $\delta = 1/6\delta^*, \delta' = 1/2\delta^*, \hat{\delta} = 1/3\delta^*$; calibrate γ to meet the privacy requirement for each λ; sample $\hat{\theta}$ using (λ, γ) and return the one with the best validation accuracy.

Figure 6.1 demonstrates how the MSE error of the linear regression algorithms varies with the privacy budget ϵ. OutPert suffers from the large global sensitivity of output θ. OPS performs well but does not benefit from the data-dependent quantities. AdaOPS is able to adaptively choose (λ, γ) based on the dataset, but suffers from the estimation error of the data-dependent empirical risk. On the other hand, OPS-PTR selects a (λ, γ) pair that minimizes the empirical error on the validation set directly, and the privacy parameter γ adapts to the dataset thus achieving the best result.

6.4.6 PATE

In this section, we apply generalized PTR to solve an open problem from Private Aggregation of Teacher Ensembles (PATE) [Papernot et al., 2017, 2018] — privately publishing the entire model through sanitizing the data-dependent DP losses. Our algorithm uses of smooth sensitivity [Nissim et al., 2007] and the Gaussian mechanism to construct a high-probability test of the data-dependent DP. Data-dependent DP is one-dimensional, enabling efficient computation under the smooth sensitivity framework. This approach is thus generally applicable for private data-adaptive analyses beyond PATE.

PATE is a knowledge transfer framework for model-agnostic private learning. In this framework, an ensemble of teacher models is trained on the disjoint private data and uses the teachers' aggregated consensus answers to supervise the training of a "student" model agnostic to the underlying machine-learning algorithms. By publishing only the aggregated answers and by the careful analysis of the "consensus", PATE has become a practical technique in recent private model training.

The tight privacy guarantee of PATE heavily relies on a delicate data-dependent DP analysis, for which the authors of PATE use the smooth sensitivity framework to privately publish the data-dependent privacy cost. However, it remains an open problem to show that the released model is DP under data-dependent analysis. Our generalized PTR resolves this gap by carefully testing a private upper bound of the data-dependent privacy cost. Our algorithm is fully described in Algorithm 9, where the modification over the original PATE framework is highlighted in blue.

Algorithm 9 takes the input of privacy budget $(\epsilon', \hat{\epsilon}, \delta)$, unlabeled public data $x_{1:T}$ and K teachers' predictions on these data. The parameter ϵ denotes the privacy cost of publishing the data-dependent DP and ϵ' is the predefined privacy budget for testing. $n_j(x_i)$ denotes the the number of teachers that agree on label j for x_i and C denotes the number of classes. The goal is to privately release a list of plurality outcomes — $\text{argmax}_{j\in[C]} n_j(x_i)$ for $i \in [T]$ — and use these outcomes to supervise the training of a "student" model in the public domain. The parameter σ_1 denotes the noise scale for the vote count.

In their privacy analysis, Papernot et al. [2018] compute the data-dependent $\text{RDP}_{\sigma_1}(\alpha, X)$ of labeling the entire group of student queries. $\text{RDP}_{\sigma_1}(\alpha, X)$ can be orders of magnitude smaller than its data-independent version if there is a strong agreement among teachers. Note that $\text{RDP}_{\sigma_1}(\alpha, X)$ is a function of the RDP order α and the dataset X, analogous to our Definition 6.4.1 but subject to RDP [Mironov, 2017].

Theorem 6.4.11 ([Papernot et al., 2018]). *If the top three vote counts of x_i are $n_1 > n_2 > n_3$ and $n_1 - n_2, n_2 - n_3 \gg \sigma_1$, then the data-dependent RDP of releasing $\text{argmax}_j\{n_j + \mathcal{N}(0, \sigma_1^2)\}$ satisfies $(\alpha, \exp\{-2\alpha/\sigma_1^2\}/\alpha)$-RDP and the data-independent RDP (using the*

Algorithm 9 PATE with generalized PTR

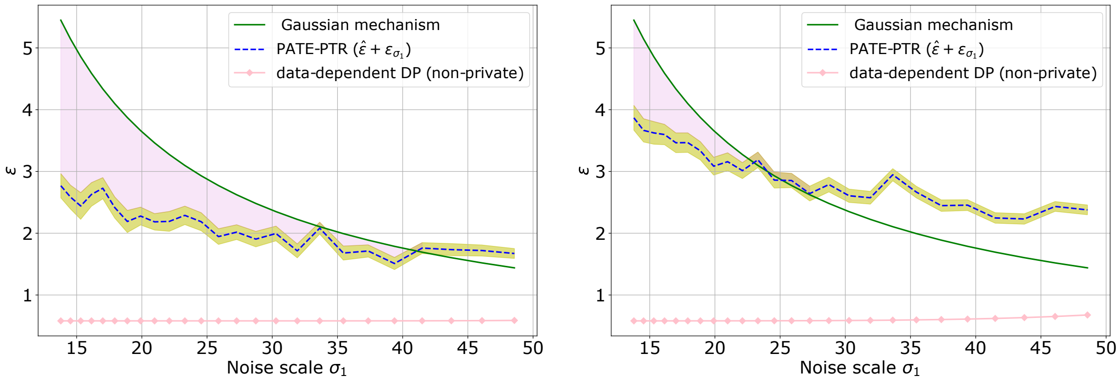
```
**Input**: Unlabeled public data $x_{1:T}$, aggregated teachers prediction $n(\cdot)$, privacy parameter $\hat{\epsilon}, \epsilon', \delta$, noisy parameter $\sigma_1$.
Set $\alpha = \frac{2\log(2/\delta)}{\hat{\epsilon}} + 1$, $\sigma_s = \sigma_2 = \sqrt{\frac{3\alpha+2}{\hat{\epsilon}}}$, $\delta_2 = \delta/2$, smoothness parameter $\beta = \frac{0.2}{\alpha}$.
Compute noisy labels: $y_i{}^p \leftarrow \text{argmax}_{j \in [C]}\{n_j(x_i) + \mathcal{N}(0, \sigma_1^2)\}$ for all $i \in [1:T]$.
$\text{RDP}_{\sigma_1}(\alpha, X) \leftarrow$ data-dependent RDP at the $\alpha$-th order.
$SS_\beta(X) \leftarrow$ the smooth sensitivity of $\text{RDP}^{\text{upper}}_{\sigma_1}(\alpha, X)$.
Privately release $\mu := \log(SS_\beta(X)) + \beta \cdot \mathcal{N}(0, \sigma_2^2) + \sqrt{2\log(2/\delta_2)} \cdot \sigma_2 \cdot \beta$
$\text{RDP}^{\text{upper}}_{\sigma_1}(\alpha) \leftarrow$ an upper bound of data-dependent RDP through Lemma 6.4.12.
$\epsilon_{\sigma_1} \leftarrow$ DP guarantee converted from $\text{RDP}^{\text{upper}}_{\sigma_1}(\alpha)$.
If $\epsilon' \geq \epsilon_{\sigma_1}$ **return** a student model trained using $(x_{1:T}; y^p_{1:T})$.
Else return $\perp$.
```

Gaussian mechanism) satisfies $(\alpha, \frac{\alpha}{\sigma_1^2})$*-RDP.*

However, $\text{RDP}_{\sigma_1}(\alpha, X)$ is data-dependent and thus cannot be revealed. The authors therefore privately publish the data-dependent RDP using the smooth sensitivity framework [Nissim et al., 2007]. The smooth sensitivity calculates a smooth upper bound on the local sensitivity of $\text{RDP}_{\sigma_1}(\alpha, X)$, denoted as $SS_\beta(X)$, such that $SS_\beta(X) \leq e^\beta SS_\beta(X')$ for any neighboring dataset X and X'. By adding Gaussian noise scaled by the smooth sensitivity (i.e., releasing $\epsilon_{\sigma_1}(\alpha, X) + SS_\beta(X) \cdot \mathcal{N}(0, \sigma_s^2)$), the privacy cost can be safely published.

Unlike most noise-adding mechanisms, the standard deviation σ_s cannot be published since $SS_\beta(X)$ is a data-dependent quantity. Moreover, this approach fails to provide a valid privacy guarantee of the noisy labels obtained through the PATE algorithm, as the published privacy cost could be smaller than the real privacy cost. Our solution in Algorithm 9 looks like the following:

- Privately release an upper bound of the smooth sensitivity $SS_\beta(X)$ with e^μ.
- Conditioned on a high-probability event of e^μ, publish the data-dependent RDP with $\text{RDP}^{\text{upper}}_{\sigma_1}(\alpha)$.



(a) High consensus and strong data-dependent DP

(b) Low consensus and low data-dependent DP

Figure 6.2: Privacy and utility tradeoffs with PATE. When σ_1 is aligned, three algorithms provide the same utility. y-axis plots the privacy cost of labeling $T = 200$ public data with $\delta = 10^{-5}$. The left figure considers the high-consensus case, where the data-adaptive analysis is preferred.

- Convert $\mathrm{RDP}^{\mathrm{upper}}_{\sigma_1}(\alpha)$ back to the standard DP guarantee using RDP to DP conversion at $\delta/2$.
- Test if the converted DP is above the predefined budget ϵ'.

The following lemma states that $\mathrm{RDP}^{\mathrm{upper}}_{\sigma_1}(\alpha)$ is a valid upper bound of the data-dependent RDP.

Lemma 6.4.12 (Private upper bound of data-dependent RDP)**.** *We are given a RDP function* $\mathrm{RDP}(\alpha, X)$ *and a β-smooth sensitivity bound $SS(\cdot)$ of* $\mathrm{RDP}(\alpha, X)$*. Let μ (defined in Algorithm 9) denote the private release of* $\log(SS_\beta(X))$*. Let the $(\beta, \sigma_s, \sigma_2)$-GNSS mechanism be*

$$\mathrm{RDP}^{upper}(\alpha){:=}\mathrm{RDP}(\alpha{,}X){+}SS_\beta(X){\cdot}\mathcal{N}(0{,}\sigma_s^2){+}\sigma_s\sqrt{2\log(\tfrac{2}{\delta_2})}e^{\mu}$$

Then, the release of $\mathrm{RDP}^{upper}(X)$ *satisfies $(\alpha, \frac{3\alpha+2}{2\sigma_s^2})$-RDP for all $1 < \alpha < \frac{1}{2\beta}$; w.p. at least $1 - \delta_2$,* $\mathrm{RDP}^{upper}(\alpha)$ *is an upper bound of* $\mathrm{RDP}(\alpha, X)$*.*

The proof (deferred to the appendix) makes use of the facts that: (1) the log of

$SS_\beta(X)$ has a bounded global sensitivity β through the definition of smooth sensitivity; (2) releasing $\mathrm{RDP}_{\sigma_1}(\alpha, X) + SS_\beta(X) \cdot \mathcal{N}(0, \sigma_s^2)$ is $(\alpha, \frac{\alpha+1}{\sigma_s^2})$-RDP (Theorem 23 from Papernot et al. [2018]).

Now we can state the privacy guarantee of Algorithm 9.

Theorem 6.4.13. *Algorithm 9 satisfies $(\epsilon' + \hat{\epsilon}, \delta)$-DP.*

In the proof, the choice of α ensures that the cost of the $\delta/2$ contribution (used in the RDP-to-DP conversion) is roughly $\hat{\epsilon}/2$. Then the release of $\mathrm{RDP}^{\text{upper}}_{\sigma_1}(\alpha)$ with $\sigma_s = \sqrt{\frac{2+3\alpha}{\hat{\epsilon}}}$ accounts for another cost of $(\epsilon/2, \delta/2)$-DP.

Empirical results. We next empirically evaluate Algorithm 9 (PATE-PTR) on the MNIST dataset. Following the experimental setup from Papernot et al. [2018], we consider the training set to be the private domain, and the testing set is used as the public domain. We first partition the training set into 400 disjoint sets and 400 teacher models, each trained individually. Then we select $T = 200$ unlabeled data from the public domain, with the goal of privately labeling them. To illustrate the behaviors of algorithms under various data distributions, we consider two settings of unlabeled data, high-consensus and low-consensus. In the low-consensus setting, we choose T unlabeled data such that there is no high agreement among teachers, so the advantage of data-adaptive analysis is diminished. We provide further details on the distribution of these two settings in the appendix.

Baselines. We consider the Gaussian mechanism as a data-independent baseline, where the privacy guarantee is valid but does not take advantage of the properties of the dataset. The data-dependent DP (Papernot et al. [2018]) serves as a non-private baseline, which requires further sanitation. Note that these two baselines provide different privacy analyses of the same algorithm (see Theorem 6.4.11).

Figure 6.2 plots privacy-utility tradeoffs between the three approaches by varying the noise scale σ_1. The purple region denotes a set of privacy budget choices ($\hat{\epsilon} + \epsilon'$ used in Algorithm 9) such that the utility of the three algorithms is aligned under the same σ_1. In more detail, the purple region is lower-bounded by $\hat{\epsilon} + \epsilon_{\sigma_1}$. We first fix $\sigma_s = \sigma_2 = 15$ such that $\hat{\epsilon}$ is fixed. Then we empirically calculate the average of ϵ_{σ_1} (the private upper bound of the data-dependent DP) over 10 trials. Running Algorithm 9 with any choice of $\hat{\epsilon} + \epsilon'$ chosen from the purple region implies $\epsilon' > \epsilon_{\sigma_1}$. Therefore, PATE-PTR will output the same noisy labels (with high probability) as the two baselines.

Observation As σ_1 increases, the privacy loss of the Gaussian mechanism decreases, while the data-dependent DP curve does not change much. This is because the data-dependent DP of each query is a complex function of both the noise scale and the data and does not monotonically decrease when σ_1 increases. However, the data-dependent DP still dominates the Gaussian mechanism for a wide range of σ_1. Moreover, PATE-PTR nicely interpolates between the data-independent DP guarantee and the non-private data-adaptive DP guarantee. In the low-consensus case, the gap between the data-dependent DP and the DP guarantee of the Gaussian mechanism unsurprisingly decreases. Meanwhile, PATE-PTR (the purple region) performs well when the noise scale is small but deteriorates when the data-independent approach proves more advantageous. This example demonstrates that using PTR as a post-processing step to convert the data-dependent DP to standard DP is effective when the data-adaptive approach dominates others.

6.5 Omitted examples and proofs

In this section, we provide more examples to demonstrate the merits of generalized PTR. We focus on a simple example of post-processed Laplace mechanism in Section 6.5 and then an example on differentially private learning of generalized linear models in Section 6.4. In both cases, we observe that generalized PTR provides data-adaptive algorithms with formal DP guarantees that are simple, effective and not previously proposed in the literature (to the best of our knowledge).

Limits of the classic PTR in private binary voting

The following example demonstrates that classic PTR does not capture sufficient data-dependent quantities even when the local sensitivity exists and can be efficiently tested.

Example 6.5.1. Consider a binary class voting problem: n users vote for a binary class $\{0, 1\}$ and the goal is to output the class that is supported by the majority. Let n_i denote the number of people who vote for the class i. We consider the report-noisy-max mechanism:

$$\mathcal{M}(X) : \text{argmax}_{i\in[0,1]} n_i(X) + Lap(b),$$

where $b = 1/\epsilon$ denotes the scale of Laplace noise.

In the example, we will (1) demonstrate the merit of data-dependent DP; and (2) empirically compare classic PTR with generalized PTR.

We first explicitly state the data-dependent DP.

Theorem 6.5.2. *The data-dependent DP of the above example is*

$$\epsilon(X) := \max_{X'}\{|\log\frac{p}{p'}|, |\log\frac{1-p}{1-p'}|\},$$

where $p := \Pr[n_0(X) + Lap(1/\epsilon) > n_1(X) + Lap(1/\epsilon)]$ *and* $p' := \Pr[n_0(X') + Lap(1/\epsilon) > n_1(X')+Lap(1/\epsilon)]$*. There are four possible neighboring datasets* $X' : n_0(X') = \max(n_0(X)\pm 1, 0), n_1(X') = n_1(X)$ *or* $n_0(X') = n_0(X), n_1(X') = \max(n_1(X) \pm 1, 0)$.

In Figure 6.3(a), we empirically compare the above data-dependent DP with the Laplace mechanism by varying the gap between the two vote counts $|n_0(X) - n_1(X)|$. The noise scale is fixed to $\epsilon = 10$. The data-dependent DP substantially improves over the standard DP if the gap is large. However, the data-dependent DP is a function of the dataset. We next demonstrate how to apply generalized PTR to exploit the data-dependent DP.

Notice that the probability $n_0(X) + Lap(1/\epsilon) > n_1(X) + Lap(1/\epsilon)$ is equal to the probability that a random variable $Z := X - Y$ exceeds $\epsilon(n_1(X) - n_0(X))$, where X, Y are two independent Lap(1) distributions. We can compute the pdf of Z through the convolution of two Laplace distributions, which implies $f_{X-Y}(z) = \frac{1+|z|}{4e^{|z|}}$. Let t denote the difference between $n_1(X)$ and $n_0(X)$, i.e., $t = n_1(X) - n_0(X)$. Then we have

$$p = \Pr[Z > \epsilon \cdot t] = \frac{2 + \epsilon \cdot t}{4\exp(\epsilon \cdot t)}$$

Similarly, $p' = \frac{2 + \epsilon \cdot (t + \ell)}{4\exp(\epsilon \cdot (t + \ell))}$, where $\ell \in [-1, 1]$ denotes adding or removing one data point to construct the neighboring dataset X'. Therefore, we can upper bound $\log(p/p')$ by

$$\begin{aligned}
\log \frac{p}{p'} &= \frac{2 + \epsilon \cdot t}{4\exp(\epsilon \cdot t)} \cdot \frac{4\exp(\epsilon(t + \ell))}{2 + \epsilon \cdot (t + \ell)} \\
&\leq \epsilon \cdot \log \left(\frac{2 + \epsilon t}{2 + \epsilon(t + 1)} \right) \\
&= \epsilon \log \left(1 - \frac{\epsilon}{2 + \epsilon(t + 1)} \right)
\end{aligned}$$

Then we can apply generalized PTR by privately lower-bounding t.

On the other hand, the local sensitivity $\Delta_{LS}(X)$ of this noise-adding mechanism is 0 if $t > 1$. Specifically, if the gap is larger than one, adding or removing one user will not change the result. To apply classic PTR, we let $\gamma(X)$ denote the distance to the nearest dataset X'' such that $\Delta_{LS} > 0$ and test if $\gamma(X) + \text{Lap}(1/\epsilon) > \frac{\log(1/\delta)}{\epsilon}$. Notice in this example that $\gamma(X) = \max(t-1, 0)$ can be computed efficiently. We provide the detailed implementation of these approaches.

1. Gen PTR: lower bound t with $t^p = t - \frac{log(1/\delta)}{\tilde{\epsilon}} + \text{Lap}(1/\tilde{\epsilon})$. Calculate an upper bound of data-dependent DP ϵ^p using Theorem 6.5.2 with t^p. The algorithm then tests if ϵ^p is within an predefined privacy budget ϵ'. If the test passes, the algorithm returns $\text{argmax}_{i\in[0,1]} n_i(X) + Lap(1/\epsilon)$ satisfies $(\tilde{\epsilon} + \epsilon', \delta)$-DP.

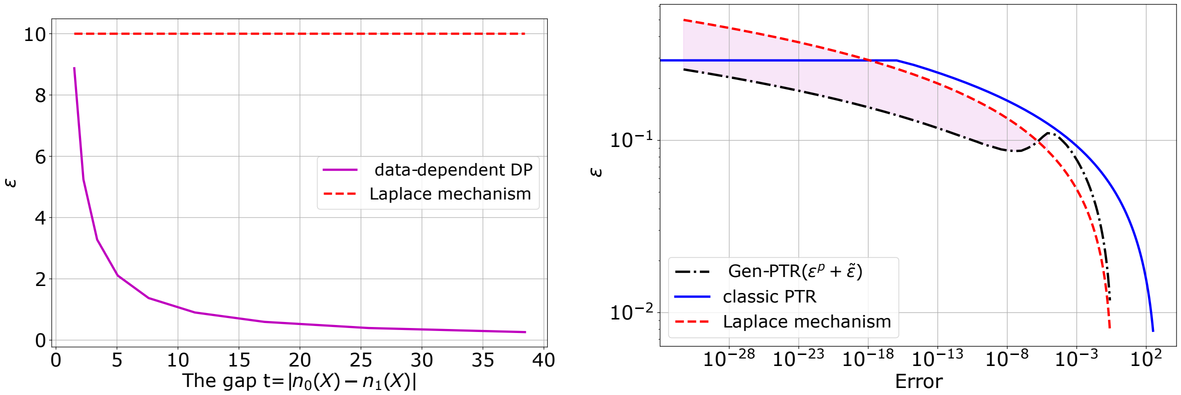
2. classic PTR: lower bound t with $t^p = t - \frac{log(1/\delta)}{\tilde{\epsilon}} + \text{Lap}(1/\tilde{\epsilon})$. If $t^p > 1$, classic PTR outputs the ground-truth result else returns a random class. This algorithm satisfies $(\tilde{\epsilon}, \delta)$-DP.

3. Laplace mechanism. $\mathcal{M}(X) : \text{argmax}_{i\in[0,1]} n_i(X) + Lap(1/\epsilon)$. $\mathcal{M}$ is (ϵ, δ)-DP.

We argue that though the Gen-PTR and the classic PTR are similar in privately lower-bounding the data-dependent quantity t, the latter does not capture sufficient information for data-adaptive analysis. That is to say, only testing the local sensitivity restricts us from learning helpful information to amplify the privacy guarantee if the test fails. In contrast, our generalized PTR, where privacy parameters and the local sensitivity parameterize the data-dependent DP, can handle those failure cases nicely.

To confirm this conjecture, Figure 6.3(b) plots a privacy-utility trade-off curve between these three approaches. We consider a voting example with $n_0(X) = n_1(X) + 100$ and $t = 100$, chosen such that the data-adaptive analysis is favorable.

In Figure 6.3(b), we vary the noise scale $b = 1/\epsilon$ between $[0, 0.5]$. For each choice of b, we plot the privacy guarantee of three algorithms when the error rate is aligned. For Gen-PTR, we set $\tilde{\epsilon} = \frac{1}{2b}$ and empirically calculate ϵ^p over 100000 trials.



(a) data-dependent DP vs Laplace mechanism

(b) Privacy-utility tradeoff between three approaches.

Figure 6.3: In Figure 6.3(a), we compare the privacy guarantee by varying the gap. In Figure 6.3(b) We fix $t = n_0(X) - n_1(X) = 100$ and compare privacy cost when the accuracy is aligned. Gen-PTR with any choice of privacy budget $(\tilde{\epsilon} + \epsilon')$ chosen from the purple region would achieve the same utility as Laplace mechanism but with a smaller privacy cost. The curve of Gen-PTR is always below than that of the classic PTR, which implies that Gen-PTR can result a tighter privacy analysis when the utility is aligned.

In the plot, when $\epsilon \ll \frac{\log(1/\delta)}{t}$, the classic PTR is even worse than the Laplace mechanism. This is because the classic PTR is likely to return $\perp$ while the Laplace mechanism returns $\mathrm{argmax}_{i\in[0,1]} n_i(X) + \mathrm{Lap}(1/\epsilon)$, which contains more useful information. Compared to the Laplace mechanism, Gen-PTR requires an extra privacy allocation $\tilde{\epsilon}$ to release the gap t. However, it still achieves an overall smaller privacy cost when the error rate $\leq 10^{-5}$ (the purple region). Meanwhile, Gen-PTR dominates the classic PTR (i.e., the dashed black curve is always below the blue curve). Note that the classic PTR and the Gen-PTR utilize the gap information differently: the classic PTR outputs $\perp$ if the gap is not sufficiently large, while the Gen-PTR encodes the gap into the data-dependent DP function and tests the data-dependent DP in the end. This empirical result suggests that testing the local sensitivity can be loosely compared to testing the data-dependent DP. Thus, Gen-PTR could provide a better privacy-utility trade-off.

6.5.1 Other applications of generalized PTR

Besides one-posterior sampling for GLMs, there are plenty of examples that our generalized-PTR could be applied, e.g., DP-PCA [Dwork et al., 2014c] and Sparse-DP-ERM [Kifer et al., 2012] (when the designed matrix is well-behaved).

[Dwork et al., 2014c] provides a PTR style privacy-preserving principle component analysis (PCA). The key observation of [Dwork et al., 2014c] is that the local sensitivity is quite "small" if there is a large eigengap between the k-th and the $k+1$-th eigenvalues. Therefore, their approach (Algorithm 2) chooses to privately release a lower bound of the k-th eigengap (k is fixed as an input) and use that to construct a high-confidence upper bound of the local sensitivity.

For noise-adding mechanisms, the local sensitivity is proportional to the data-dependent loss and generalized PTR is applicable. We can formulate the data-dependent DP of DP-PCA as follows:

Theorem 6.5.3. *For a given matrix $A \in \mathcal{R}^{m\times n}$, assume each row of A has a bounded ℓ_2 norm being 1. Let V_k denotes the top k eigenvectors of A^TA and d_k denotes the gap between the k-th and the $k+1$-th eigenvalue. Then releasing $V_kV_k^T + E$, where $E \in \mathcal{R}^{n\times n}$ is a symmetric matrix with the upper triangle is i.i.d samples from $\mathcal{N}(0,\sigma^2)$ satisfies $(\epsilon(A),\delta)$ data-dependent DP and $\epsilon(A) = \frac{2\sqrt{\log(1.25/\delta)}}{\sigma(d_k-2)}$.*

The proof is based on the local sensitivity result from [Dwork et al., 2014c] and the noise calibration of Gaussian mechanism.

We can combine Theorem 6.5.3 with our Algorithm 8 to instantiate the generalized PTR framework. The improvement over Dwork et al. [2014c] will be to allow joint tuning of the parameter k and the noise variance (added to the spectral gap d_k).

Chapter 7

Sparse vector technique (SVT)

7.1 Introduction

The Sparse Vector Technique (SVT) [Dwork et al., 2009] is a fundamental tool in differential privacy (DP) that allows the algorithm to screen potentially an unbounded number of adaptively chosen queries while paying a cost of privacy only for a small number of queries that passes a predefined threshold.

SVT is the workhorse behind the *private multiplicative weights* mechanism [Hardt and Rothblum, 2010] and *median oracle* mechanism [Roth and Roughgarden, 2010], which famously shows that one can answer exponentially more linear queries differential privately for low-dimensional problems. It is also the key technique underlying the (conjectured optimal) improvements to the *ReusableHoldout* algorithms for preserving statistical validity in *adaptive data analysis* [Dwork et al., 2014a] and the *Ladder* algorithm for reliable machine learning leaderboards [Blum and Hardt, 2015]. We refer readers to the excellent course [Smith and Roth, 2017, Lecture 12] and the references therein.

More recently, SVT is combined with the *Distance to Stability* argument to build a machinery for *model agnostic private learning* in the knowledge transfer framework

[Bassily et al., 2018]. The proposed algorithm releases many private labels from an ensemble of "teacher" classifiers trained on the private dataset [Bassily et al., 2018] while essentially only paying a privacy cost for those that are unstable. This in principle would allow the use any deep neural networks as a blackbox while leveraging the high-margin of the learned representation.

Despite the substantial benefit of SVT in theory, it is not known as a practical method. For example, in the case of model-agnostic private learning, SVT is often outperformed by simple Gaussian Mechanism [Papernot et al., 2018] that release all labels, since the latter uses a more concentrated noise (Gaussian over Laplace) and also has a tighter composition via Concentrated / Renyi differential privacy (CDP/RDP) [Dwork and Rothblum, 2016, Bun and Steinke, 2016, Mironov, 2017].

In this paper, we revisit SVT and address the following questions:

1. Is it essential to add Laplace noise? Does Gaussian noise work too? How about other noises e.g., [Geng and Viswanath, 2014]?

2. Is there a tighter RDP bound for SVT? Can we parameterize the RDP of SVT by the RDP function of the randomized mechanisms that are used to perturb the threshold and the answer to each query?

3. So far, the advanced composition of SVT is only available for the case when we compose c SVTs with cut-off $= 1$, which requires refreshing the threshold noise each time. Could there be an $\sqrt{c}$ composition-theorem for the more general version when $c > 1$?

4. Finally, can we achieve better utility of SVT in practice? How small does c needs to be relative to the total number of queries k before SVT can outperform naive Gaussian mechanism?

5. Are there more practical alternatives to SVT that operates in those regimes where SVT fails.

We answer affirmatively to the first three questions (with some caveats and restrictions) by studying a generalized family of SVT (see Algorithm 11). Then we conduct numerical experiments to illustrate the pros and cons of various algorithms while highlighting the challenges in the last two questions. Moreover, we applied our results to the problem of adaptive data analysis and provided a "high probability" bound on the maximum accuracy of a sequence of k adaptively chosen queries based on a Gaussian-mechanism variant of SVT, which matches (but unfortunately not improving) the strongest bound known to date on this problem.

A remark on our novelty. We believe our technical analysis that derives the RDP bound is new and elegant. Also our empirical evaluation is by far the more extensive for SVT-like algorithms. That said, we do borrow ideas from various prior work including [Lyu et al., 2017, Smith and Roth, 2017, Hardt and Rothblum, 2010] for the analysis including a cute trick from [Bun and Steinke, 2016], as well as getting practical insight and inspiration from [Papernot et al., 2018]'s data-dependent analysis of *noisy-screening*. A recent work [Liu and Talwar, 2019] generalized SVT to beyond low-sensitivity queries but still uses Laplace noise. We are different in that we develop SVT with other noise-adding mechanisms. Our technique should be directly applicable to the BetweenThreshold variant as in [Bun et al., 2017] an also release the "gap" as in [Ding et al., 2019]. The overarching goal of the paper is to make progress in bringing an amazing theoretical tool to practice. The improvements might be a constant factor in certain regimes but as differential privacy transitions into a practical technology, "constant matters!"

Symbols and notations. Throughout the paper, we will use standard notations for probability unless otherwise stated, e.g., $\Pr[\cdot]$ for probability, $p[\cdot]$ for density, $\mathbb{E}[\cdot]$ for expectation. Conditional probabilities, density and expectations are denoted with the standard | in the middle, e.g., $\mathbb{E}[\cdot|\cdot]$, except for the cases when we state upfront that they abbreviated for lighter notations in that section. We do not distinguish fixed parameters and random variables as they are clear from context. The randomness are entirely the randomness induced by the randomized algorithm, except in the last section when we talk about adaptive data analysis. ϵ, δ are reserved for privacy budget/loss parameters, and α the order of RDP. Other notations will be defined on the fly as they first appear.

The most common mechanisms for differential privacy are those that add noise to queries answers.

Definition 7.1.1 (Noise-adding mechanisms)**.** We say that $\mathcal{M} : \text{Data} \times \mathcal{Q} \rightarrow P_{\mathbb{R}}$ is a noise-adding mechanism if it answers a query q by outputting $o \sim \mathcal{M}(D, q) = q(D) + Z$ where Z is a random variable.

Typical examples of these noise-adding mechanisms for differential privacy includes Laplace-mechanism, Gaussian mechanism in which Z is drawn from a Laplace distribution and a Gaussian distribution respectively. Notably, the "optimal" geometric mechanism falls under this category which adds a "stair-case"-shape noise [Geng and Viswanath, 2014].

Definition 7.1.2 (Low-sensitivity queries)**.** We define $\mathcal{Q}(\triangle)$ to be the set of all queries $q : \text{Data} \rightarrow \mathbb{R}$ such that $|q(D) - q(D')| \leq \triangle$ for any pair of neighboring datasets D, D'.

$\triangle$ is called global sensitivity and is used to calibrate the noise according to a given privacy budget.

7.1.1 Sparse vector techniques

In SVT, the input is a stream of possibly infinitely long, adaptively chosen queries $q_1, q_2, ..., q_i, ... \in \mathcal{Q}(\triangle)$. The queries are provided with a sequence of thresholds $T_1, T_2, ..., T_k, ...$

Algorithm 10 Standard SVT

Input: Data D, an adaptive sequence of queries $q_1, q_2, ... \in \mathcal{Q}$ with sensitivity $\triangle$, privacy parameter ϵ_1, ϵ_2, threshold T, cut-off c, option $\mathsf{RESAMPLE}$.

```
Sample $\rho \sim \mathsf{Lap}(\triangle/\epsilon_1)$, count $= 0$
for $i = 1, 2, 3, ...$
Sample $\nu_i \sim \mathsf{Lap}(2\triangle/\epsilon_2)$
if $q_i(D) + \nu_i \geq T_i + \rho$ then
    Output $a_i = \top$, count $=$ count$+1$
    if RESAMPLE, $\rho \sim \mathsf{Lap}(\triangle/\epsilon_1)$.
    if count $\geq c$, abort.
else
    Output: $a_i = \perp$
end if
```

Algorithm 11 Generalized SVT

Input: Data D, an adaptive sequence of queries $q_1, q_2, ... \in \mathcal{Q}$ with sensitivity $\triangle$, noise-adding mechanisms $\mathcal{M}_\rho, \mathcal{M}_\nu$, threshold T, cut-off c, max-length $k_{\max}$, option $\mathsf{RESAMPLE}$.

```
Sample $\hat{T} \sim \mathcal{M}_\rho(D, T)$, count $= 0$
for $i = 1, 2, 3, ..., k_{\max}$
Sample $\hat{q}_i \sim \mathcal{M}_\nu(D, q_i)$
if $\hat{q}_i \geq \hat{T}$ then
    Output $a_i = \top$, count $=$ count $+ 1$
    if RESAMPLE, $\hat{T} \sim \mathcal{M}_\rho(D, T)$
    if count $\geq c$, abort.
else
    Output: $a_i = \perp$
end if
```

The goal of SVT is to release a binary vector $\{\perp, \top\}^k$ at every time k, $\top$ indicates that the corresponding query answer $q_i(D)$ is above the threshold T_i and $\perp$ indicates below. To release this vector differential privately, we first perturb the threshold T with a Laplace noise ρ. Then each individual query $q_i(D)$ is perturbed by another Laplace noise ν_i before comparing against the perturbed threshold $T + \rho$ to determine the binary decion, until the stopping condition — the c-th $\top$ arrives. Algorithm 10 summarizes pseudo-code from [Hardt and Rothblum, 2010] and [Lyu et al., 2017].

A remarkable property of SVT is that it allows the release of a vector that is exponentially long while incurring only a privacy loss proportional to c (or its square root) — the maximum number of answers that are allowed to be $\top$. This is formalized in the following lemma.

Lemma 7.1.3 (Privacy calibration in Standard SVT)**.** *Algorithm 10 satisfies* $(\epsilon_1 + c\epsilon_2)$*-DP when* $\mathsf{RESAMPLE}$ *option is set to false. When* $\mathsf{RESAMPLE} = \text{True}$*, then Algorithm 10 with* $\frac{\triangle}{\epsilon_1} = \frac{\triangle}{\epsilon_2} = \frac{\sqrt{32c\log(1/\delta)}}{\epsilon}$*, then Algorithm 10 satisfies* (ϵ, δ)*-DP and* $(c\epsilon_1 + c\epsilon_2)$*-DP.*

The version of the pure-DP calibration without resampling comes from [Lyu et al., 2017, Algorithm 1]. The (ϵ, δ)-DP calibration is extracted from [Dwork and Roth, 2013, Theorem 3.25], which is essentially applying strong composition to to c instances of SVT, each obeying $(\epsilon_1 + \epsilon_2)$-DP.

Despite the asymptotic savings from $\sqrt{k}$ to $\sqrt{c}$, SVT is still not known as a practical mechanism. The reasons, in our opinion, are twofolds.

The Laplace distribution used in the SVT is a heavy-tailed (sub-exponential) distribution, which requires the threshold to be set to $O(\log(1/\beta))$ so as to control the false positive rate at β. This could be much larger than the $O(\sqrt{\log(1/\beta)})$ of sub-gaussian tailed distributions, hence make SVT less favorable for utility-privacy trade-off in practice. Moreover, many practical differential private algorithms benefit from tighter privacy accounting, e.g., composition using Renyi DP with numerical computation. It will be ideal if we can come up with a version of the SVT that adds more concentrated noise as well as a general Renyi DP analysis of that algorithm. This motivated us to consider the family of Generalized SVT mechanism in Algorithm 11.

7.2 Main results: RDP bounds for SVT variants

In this section, we derive RDP bounds for SVT variants with different distributions of noisy parameters (ν_i, ρ). The goal is to find those distributions that not only have thin tail bounds but preserve the essential property of the standard SVT — they can answer exponentially many $\perp$ queries while paying a privacy loss only proportional to c or $\sqrt{c}$ when the algorithm halts. The family of mechanisms we consider is summarized in Algorithm 11. The differences from Algorithm 10 are highlighted in blue.

7.2.1 RDP analysis with $c = 1$

We first consider generalized SVT with $c = 1$, since the case of $c > 1$ is often treated as composition of multiple SVT with $c = 1$.

Theorem 7.2.1. *Let K be a random variable indicating the stopping time — number of $\perp$s plus 1. Let $\mathcal{M}_\rho$, $\mathcal{M}_\nu$ be noise-adding mechanisms (Definition 7.1.1). Assume $\mathcal{M}_\rho$ satisfies $\epsilon_\rho(\alpha)$-RDP for queries with sensitivity $\triangle$ and $\mathcal{M}_\nu$ satisfies $\epsilon_\nu(\alpha)$-RDP for queries with sensitivity $2\triangle$. Then Algorithm 11 with $c = 1$ (denoted by $\mathcal{M}$) obeys*

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \epsilon_\rho(\alpha) + \epsilon_\nu(\alpha) + \frac{\log \sup_z \mathbb{E}[K|\rho = z]}{\alpha - 1}, \tag{7.1}$$

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \frac{\alpha - (\gamma-1)/\gamma}{\alpha - 1}\epsilon_\rho(\frac{\gamma}{\gamma-1}\alpha) + \epsilon_\nu(\alpha) + \frac{\log\left(\mathbb{E}_{z\sim p_\rho}[\mathbb{E}[K|\rho = z]^\gamma]\right)}{\gamma(\alpha-1)}, \tag{7.2}$$

for all $\gamma > 1$ and $1 < \alpha < \infty$. Moreover, when $\epsilon_\rho(\infty) \leq \infty$, we get

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \epsilon_\rho(\alpha) + \epsilon_\nu(\infty). \tag{7.3}$$

The theorem can be thought of as a general transfer theorem that allows us to bound the RDP of the generalized SVT with the RDP of its subroutines $\mathcal{M}_\rho$ and $\mathcal{M}_\nu$. Before proving the theorem in Section 7.6.1, let us parse the result in a number of special cases.

Remark 7.2.2 (Pure-DP). RDP (7.3) recovers the pure-DP bound of the standard SVT when $\alpha \to \infty$. It also allows other noise-adding procedure that satisfies pure-DP to be applied. We could also consider the hybrid-noise SVT where ρ is a Gaussian noise, but ν are Laplace-noises.

Remark 7.2.3 (Bounded-length SVT). When we set $k_{\max} < +\infty$, the (7.1) implies an RDP bound of the form $\epsilon_\rho(\alpha) + \epsilon_\nu(\alpha) + \log(1 + k_{\max})/(\alpha - 1)$, which further implies an

(ϵ, δ)-DP bound by Lemma **??**. In particular, if $\delta \leq 1/(1+k_{\max})$, then we get (ϵ, δ)-DP with

$$\epsilon = \min_{\alpha>1} \epsilon_\rho(\alpha) + \epsilon_\nu(\alpha) + 2\log(1/\delta)/(\alpha-1).$$

In the case of Gaussian mechanism, this loses at most a factor of $\sqrt{2}$ comparing to the case when $\log(1+k_{\max})/(\alpha-1)$ is not there all together.

When $k_{\max}$ is chosen to be $+\infty$: (7.1) and (7.2) do not imply RDP in this case, because there are cases where SVT can potentially have unbounded length (in fact, even the expected length can be unbounded.). It is well-expected that if we use Gaussian-mechanism as a subroutine for SVT, the dependence of the sequence length is unavoidable. Similar observations have been made about a Gaussian-noise version of the ReportNoisyMax mechanism [see, e.g., Dwork and Roth, 2013, Section 3.5.3]. That said, the form of the bound (7.2), which depends only on the moments of the conditional expectation seems to suggest that we can potentially obtain meaningful RDP bounds for generalized SVT even if $k_{\max} = +\infty$ in some cases.

Let us consider a mild restriction to the family of queries that can be chosen, which allows us to keep the sequence length unbounded even when the noise-adding subroutines do not satisfy pure-DP.

Definition 7.2.4 (Nonnegative, Low-sensitivity Queries Model)**.** The adversary can adaptively choose $q_1, q_2, ... \in \mathcal{Q}_+(\triangle)$ where

$$\mathcal{Q}_+(\triangle) = \{q : \text{Data} \to \mathbb{R} \mid q(D) \geq 0 \forall D,\ |q(D) - q(D')| \leq \triangle \forall \text{ neighboring datasets } D, D'\}.$$

The class covers both use cases of SVT that we described earlier. When we apply SVT to "Guess-and-Check"[1], $q_i(D) = \|f_i(D) - g_i\|$ is nonnegative. Similarly, in the case

[1] A subroutine of "private multiplicative weights" and "reusable holdout".

of “Model-agnostic private learning” $q_i(D) = \mathrm{dist}_{\mathrm{MajorityVote}_{f_i}}(D)$, which measures the number of data points that need to be added or removed to make the argmax of the voting score unstable.

Proposition 7.2.5 (Gaussian SVT with non-negative queries)**.** *Let Algorithm 11 be instantiated with $\mathcal{Q}_+(\triangle)$, $\mathcal{M}_\rho$ and $\mathcal{M}_\nu$ be Gaussian mechanism with parameter σ_1 and σ_2. Then for all $T < +\infty$ and $\gamma > 1$ such that $\sigma_2 > \sqrt{\gamma+1}\sigma_1$, Algorithm 11 with $c = 1$ halts with K rounds satisfying*

$$\mathbb{E}[\mathbb{E}[K|\rho = z]^\gamma] \leq 1 + (c_\gamma\sqrt{2\pi}\max\{\frac{T(1+\gamma)}{\sigma_1}, 1\})^\gamma(1+\gamma)^{1/2}e^{\frac{\gamma T^2}{2\sigma_1^2}}.$$

For Gaussian SVT satisfying $\sigma_2 \geq \sqrt{3}\sigma_1$, it obeys an RDP of $\frac{\alpha\triangle^2}{\sigma_1^2}+\frac{2\alpha\triangle^2}{\sigma_2^2}+\frac{\log(1+2\sqrt{3}\pi(1+\frac{9T^2}{\sigma_1^2})e^{\frac{T^2}{\sigma_1^2}})}{2(\alpha-1)}$.

The proof of Proposition 7.2.5, provided in the full paper, hinges upon the key observation that K follows a Negative Binomial distribution when conditioning on the threshold, and some technical calculations involving Mill’s ratio and moments of Gaussian distribution.

Remark 7.2.6 (Controlling Type I error)*.* One can for example choose $T = \sqrt{2(\sigma_1^2+\sigma_2^2)\log 1/\varrho} = \sqrt{8\sigma_1^2\log(1/\varrho)}$ such that the Type I error (false positive rate) is bounded by ϱ. This is often the case when using sparse vector technique for statistical applications. Then we can simplify the above bound by using $\log(1+x) \leq x$, and an assumption that ρ is sufficiently small, to obtain an RDP bound of

$$\frac{5\alpha\triangle^2}{3\sigma_1^2} + \frac{8\log(1/\rho)+\log(4\sqrt{3}\pi(1+72\log(1/\rho)))}{2(\alpha-1)} \leq \frac{5\alpha\triangle^2}{3\sigma_1^2} + \frac{5\log(1/\rho)}{(\alpha-1)}.$$

The above results allow us to obtain nearly the same (ϵ,δ)-DP bound for Gaussian-SVT as if we are working with the RDP bound of a Gaussian mechanism, provided that the δ chosen such that $\log(1/\delta)$ is larger than either $\log k_{\max}$ in the length-bounded case or $O(\log(1/\rho))$ in the nonnegative query setting.

To ease our subsequent presentation, from here onwards we will use k_γ to denote a data-independent upper bound of $(\mathbb{E}[K|\rho]^\gamma])^{1/\gamma}$. Conveniently, $k_\infty = k_{\max}$ and k_1, which unifies (7.1) and (7.2). Moreover, we will use γ^* such that $1/\gamma^* + 1/\gamma = 1$.

7.2.2 Generalized SVT with $c > 1$

We now address the case when $c > 1$. A natural, and general way to deal with SVT for $c > 1$ is to simply apply composition theorems of differential privacy to c instances of SVT with cut-off parameter c set to 1. We could also directly analyze the variant of SVT with $c > 1$, where the threshold noise is not refreshed. Pros and cons of these two approaches are described in Appendix **??**.

Theorem 7.2.7 (RDP for length-capped SVT with $c > 1$). *The generalized SVT with cut-off parameter $c > 1$ and a maximum length is $k_{\max}$ obeys that*

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \epsilon_\rho(\alpha) + c\epsilon_\nu(\alpha) + \frac{1 + \log \sum_{k=0}^{c} \binom{k_{\max}}{k}}{\alpha - 1}.$$

The proof, presented in the Appendix, uses the same techniques as in the proof of Theorem 7.2.1, but we no longer get an interpretable bounds that rely on moments of $\mathbb{E}[K|\rho]$. The term in the logarithmic factor, resembles $k_{\max}$ in the sense that it counts the cardinality of the output space — binary vectors of length $k_{\max}$ with at most c $\perp$s.

Remark 7.2.8. When both noise are Gaussian, the theorem and Lemma **??** implies an (ϵ, δ)-DP with

$$\epsilon(\delta) \leq \frac{\Delta^2}{2\sigma_1^2} + \frac{2c\Delta^2}{\sigma_2^2} + \sqrt{2\left(\frac{\Delta^2}{2\sigma_1^2} + \frac{2c\Delta^2}{\sigma_2^2}\right)\left(\log(\delta^{-1}) + \log c\binom{k_{\max}}{c}\right)}$$

which recovers the $O(\sqrt{c})$ scaling when $\delta \leq c^{-1}\binom{k_{\max}}{c}^{-1}$ and saves a factor of c in σ_1.

While the restriction on δ being smaller than $k_{\max}^{-c}$ is quite limiting, we are not aware of an analysis that achieves the strong composition-like scaling in c for the version of the SVT that does not refresh the noise under any parameter configurations.

Back to (ϵ, δ)-composition. Interestingly, if we use of the strong composition for (ϵ, δ)-DP directly, we can obtain a bound with $\sqrt{c}$ scaling for a much broader set of parameters. Let us consider the following stage-wise algorithm for Generalized SVT, which resamples the threshold noise ρ after every c' rounds with a pre-specified bound $k'_{\max}$ chosen in each round. This algorithm can be viewed as a meta-algorithm that calls Algorithm 11 as a subroutine (see Algorithm 12). The idea is that we can choose c' and $k'_{\max}$ carefully according to c and δ such that for each call of SVT, the region of interests falls under the region where $\log(c'\binom{k'_{\max}}{c'})$ is comparable to $\log(1/\delta)$.

Theorem 7.2.9 (Stage-wise Length-Capped Gaussian SVT for $\mathcal{Q}(\triangle)$)**.** *Let $0 < \delta' < 1$ be a parameter. Let $\mathcal{M}$ be the instance of the Algorithm 12 invoked with cut-off c', max-length k', option* RESAMPLE $=$ False*, $\mathcal{M}_\rho, \mathcal{M}_\nu$ chosen as Gaussian mechanisms with noise parameter σ_1, σ_2 satisfying $\sigma_2 = 2\sigma_1$ and $\sigma_1 \geq 8\triangle\sqrt{c\log(1/\delta')}$. If we choose $c' \leq c$ such that $c'\binom{k'_{\max}}{c'} \leq (\delta')^{-1}$, then $\mathcal{M}$ satisfies $(\epsilon, \tilde{\delta} + \frac{c}{c'}\delta')$-DP with $\epsilon = O\left(\sqrt{\frac{c\triangle^2}{\sigma_1^2}\log(1/\delta')\log(1/\tilde{\delta})}\right)$.*

Theorem 7.2.10 (Adaptive Stage-wise Gaussian SVT for $\mathcal{Q}_+(\triangle)$)**.** *Let $\mathcal{M}$ be an instance of Algorithm 12 invoked with the same parameters as in Theorem 7.2.9, except that* RESAMPLE $=$ True *and $k_{\max} = +\infty$. Then for all c', γ such that $k_\gamma^{c'} \leq (\delta')^{-1}$, then $\mathcal{M}$ is $(\epsilon, \tilde{\delta} + \frac{c}{c'}\delta')$-DP with $\epsilon = O\left(\sqrt{\frac{c\triangle^2}{\sigma_1^2}\log(1/\delta')\log(1/\tilde{\delta})}\right)$ for all adaptively chosen sequences of queries in $\mathcal{Q}_+$.*

Remark 7.2.11 (Adaptive to c' and numerical computation). Observe that choosing RESAMPLE $=$ True makes Algorithm 12 identical to Algorithm 11 for all choices of

Algorithm 12 Stage-wise generalized SVT

```
Input: Data $D$, an adaptive sequence of queries $q_1, q_2, ... \in \mathcal{Q}$ with sensitivity $\triangle$, noise-adding mechanisms $\mathcal{M}_\rho, \mathcal{M}_\nu$, threshold $T$, total cut-off $c$, per-stage cut-off $c'$, per-stage max-length $k'_{\max}$, option RESAMPLE.
Initialize output vector to be an empty list.
for  for $\ell = 1, 2, 3, ..., \lceil c/c' \rceil$ do
    Set $\tilde{c} = c - c'(\lceil c/c' \rceil - 1)$ if $\ell = \lceil c/c' \rceil$, and $\tilde{c} = c'$ otherwise.
    Invoke Algorithm 11 with $D, T, \mathcal{M}_\rho, \mathcal{M}_\nu, \tilde{c}, k'_{\max}$, RESAMPLE and current front of the adaptive stream of queries.
    Append the new output vector from Algorithm 11 to the output.
end for
```

$c' \geq 1$. We can thus minimize the bound numerically over the parameters c', δ' to minimize the final bound, to simulate the conceptual process of which part of the composition is RDP-based and which part (ϵ, δ)-DP based. The best choice would be to make c' as large as possible so as to get the partial benefit of the savings from RDP composition over the c' steps within each stage, while not ruining the $O(\sqrt{c})$ strong composition when c is large.

Remark 7.2.12. Comparing to the likely-unachievable conjecture where the generalized SVT has an RDP of $\epsilon_\rho(\alpha) + c\epsilon_\nu(\alpha)$, which would give an $\epsilon = O(\sqrt{\frac{c\triangle^2}{\sigma_1^2}\log(1/\delta)}$, this bound is worse only by a factor of $\sqrt{\log(1/\delta)}$, and has some mild restrictions on δ. We remark that in Theorem 7.2.9 and 7.2.10 we focused on the asymptotic scaling, while in practice, we can use the optimal advanced composition due to [Kairouz et al., 2015] and search for the best parameters to give the tightest bounds.

7.3 Applications

7.3.1 Adaptive data analysis

The stage-wise length-bounded Gaussian SVT's (ϵ, δ)-DP guarantee allows us to directly apply it to the problem of adaptive data analysis that aims at preventing data dredging while still allowing an analyst to get accurate answers about a sequence of k adaptively chosen statistical queries through an interactive protocol [Dwork et al., 2014a, Smith, 2017].

Theorem 7.3.1. *With probability $\geq 1-\delta$ over the random coins of the i.i.d. data, our algorithm and other randomness coming from the interaction protocols against an arbitrary adaptive adversary, the Gaussian-SVT-based* Private-Guess-and-Check *answers k queries including at most c inaccurately guesses with generalization error at most $O(\frac{c^{1/4}\log(k/\delta)^{3/4}}{n^{1/2}})$.*

The proof combines either Theorem 7.2.9 or 7.2.10 with the high probability generalization bound of (ϵ, δ)-DP algorithms [Jung et al., 2020] as well as the Gaussian tail bound.

In comparison, the simple Gaussian mechanism guarantees an accuracy of $O(k^{\frac{1}{4}} \log \frac{k}{\delta}^{\frac{1}{2}} n^{-\frac{1}{2}})$ and the original *ReusableHoldout* gives $O(c^{1/2}\sqrt{\log(k/\delta)}n^{-1/2})$. We show that Gaussian SVT improves over these and matches the best known rate for the problem achieved by Laplace-mechanism-base SVT — an $O(\log(k/\delta)^{1/4})$ away from the lower bound. Interestingly the reason of the suboptimality is different. Laplace SVT is off due to the subexponential tail bound of Laplace R.V., while Gaussian SVT is off due to the additional $O(\log(k/\delta)^{1/2})$ factor from the strong composition. It remains an open problem how to close this gap.

7.3.2 Model-agnostic private learning

Model-agnostic private learning is another application of the sparse-vector technique. In this problem, the learner has access to a private labeled dataset and a public unlabeled dataset. The algorithm leverages a blackbox learner, e.g., a deep learning algorithm, by training one classifier on each randomly split of the private dataset. Then it privately labels the public dataset by privately releasing the majority-votes of these classifiers' predictions.

This scheme has been shown to be practical [Papernot et al., 2017, 2018] by combining simple Gaussian mechanism for differential privacy with semi-supervised learning approaches. Bassily et al. [2018] substantially improves the algorithm by showing that, under a PAC-learning framework, that one can privately release the labels for all public data points, while spending the privacy budget only for those data points where the voters are labeling incorrectly (or labeling inconsistently, to be more general).

SVT is applied to test whether each query has received an overwhelming majority from the voters by testing if distance-to-stability is sufficiently large. If so, the exact answer $f(D)$ is released with $\bot$ and if not, only $\top$ is released. Interestingly, this approach has a privacy loss that depends only on the number of $\top$s and it can be thought of as a composition of the SVT with the $(0, \delta)$-DP part of the event from the "Stability"-based argument.

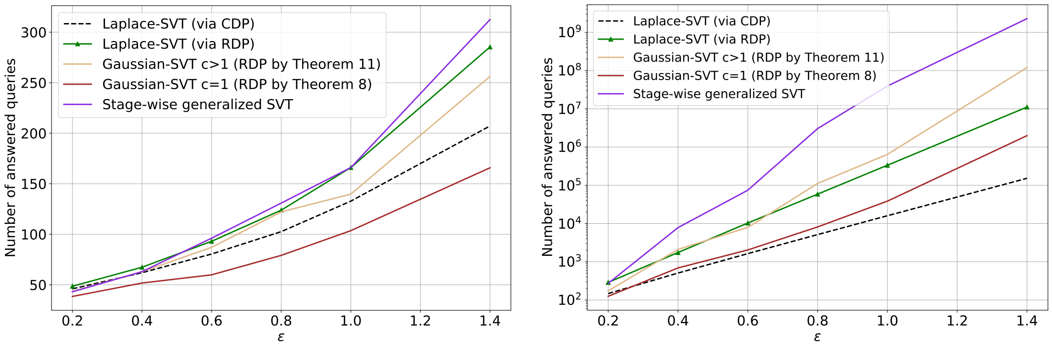
While this approach provides a substantial benefit in theory, it has been observed in practice that it is often outperformed by simple Gaussian mechanism in practice, since the latter uses a more-concentrated noise and also a much tighter composition.

In the experiment section, we demonstrate that the story is now different when Gaussian SVT is used as a drop-in replacement.

7.4 Experiment and discussion

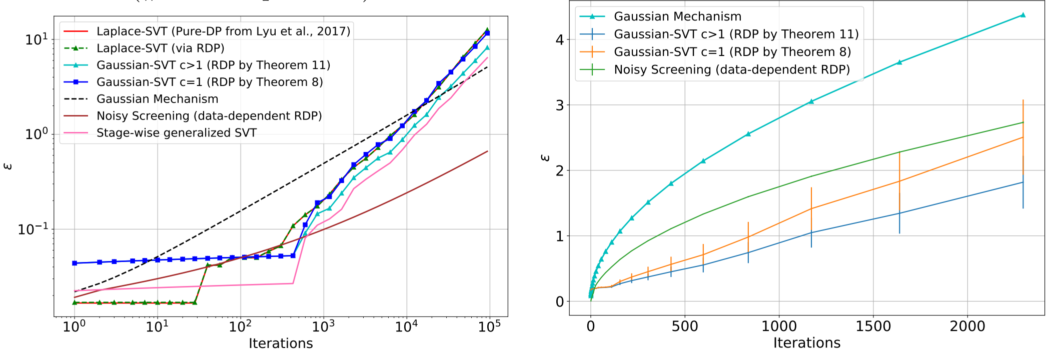
In this section, we conduct extensive numerical experiments to illustrate the behaviors of SVT variants. We will have three sets of experiments.

Exp. 1 (Calibrating noise to privacy) Given a predetermined privacy budget (ϵ, δ) and the cut-off c, we compare the length each SVT-like algorithm can screen before



(a) $T = 100, c = 20, \delta = 10^{-6}$ (b) $T = 700, c = 20, \delta = 10^{-10}$

Figure 7.1: Number of queries each algorithm can process with a fixed privacy budget (ϵ, δ), fixed cut-off (# of false positives) c and fixed threshold T.



(a) Exp 2: Synthetic null sequence (b) Exp 3: CIFAR10

Figure 7.2: Total composed privacy loss as the algorithm progresses for $\delta = 10^{-6}$. The margin $T = 1000$ and $\sigma_1 = 210$. The standard deviation of Gaussian and Laplace are aligned to be comparable.

stopping.

Exp. 2 (Privacy loss computation) We evaluate SVT variants with the same variance of noise by comparing the composed privacy loss for finishing a fixed length sequence of queries.

Exp. 3 (Real life data) We investigate various private screening methods with a realistic sequence of queries from running a kNN-based private-query release on the CIFAR-10 dataset.

In Exp 1 and Exp 2, the sequences of queries are $q_t(D) = 0$ for all t so all discoveries that end up being detected as $\top$ are false positives. Thus the length of the sequence is a measure of utility in Exp 1. Exp 2 and Exp 3 compares the expected privacy loss ϵ at a fixed δ as it composes. For all experiments, we denote (σ_1, σ_2) or (λ_1, λ_2) as the noise to perturb threshold and query in Gaussian and Laplace, respectively. The ratio between the query noise and the threshold noise is fixed — $\frac{\sigma_2}{\sigma_1} = \frac{\lambda_2}{\lambda_1} = 2$. When applicable, we include simple Gaussian mechanism as a baseline. Moreover, we added noisy-screening, which basically output $\perp$ if $q_t(D) + \mathcal{N}(0, \sigma^2) \geq T$ and $\top$ otherwise. The data-dependent RDP-bound for noisy-screening [Papernot et al., 2018, Theorem 6] behaves like SVT as it pays exponentially smaller privacy loss when the query $q_t(D)$ gets far from the threshold T. We emphasize the privacy loss is sensitive information, thus not directly comparable to other DP methods. Finally for Gaussian-SVT, $k_{\max}$ needs to be chosen carefully.

Observations on the experiments. In Experiment 1, when the tail of the noise plays a significant role, e.g. the threshold T is large (Figure 7.1(b)), Gaussian-SVT is more advantageous due to a more concentrated noise. To further improve Gaussian-SVT, the stage-wise Gaussian-SVT that uses hybrid composition (Theorem 7.2.9) outperforms Laplace-SVT significantly. On a side note, the sinh-style RDP bound for Laplace-SVT ($c = 1$) from Lemma **??** turns out to be quite a bit better than the CDP-version and the standard calibration (Lemma 7.1.3). In Experiment 2, we see that as the privacy loss composes Laplace-SVT and Gaussian-SVT with the same noise variance behave qualitatively similar. Gaussian-SVT is better by a constant factor with larger number of iterations. Meanwhile, naive Gaussian mechanism and noisy-screening is often the better choice when the number of iterations is small. In Experiment 3, we see that the expected privacy losses of Gaussian SVT outperforms that of the noisy-screening despite that the latter is data-dependent. The error bars are computed based on 10 independent run and has a correct 95% coverage. We excluded Laplace-SVT in Exp 3 due to the lack of a way for fair comparison.

7.5 Conclusion

To conclude, we developed a generalization of sparse vector technique for DP that allows us to use any noise-adding mechanisms. We derived the Renyi-DP bounds of these generalized-SVT and showed that we can get $\sqrt{c}$-composition in all practical regimes of interests. We use theory and experiments to demonstrate the merits of Gaussian-SVT. In downstream tasks, we have shown that Gaussian-SVT matches the best existing bound for adaptive data analysis and demonstrated in experiments that it could improve the privacy accounting in model-agnostic private learning. We hope the work will spark new ideas and practical applications involving SVT.

7.6 Omitted proofs

7.6.1 Proof of Theorem 7.2.1 — RDP of the Generalized SVT for $c = 1$

For unbounded sequences, the output space of the algorithm is $\{\perp^k \top | k = 0, 1, ..., \infty\}$. In the case when $k_{\max} < +\infty$, the output space is $\{\perp^k \top | k = 0, 1, ..., k_{\max} - 1\} \cup \{\perp^{k_{\max}}\}$. For notation convenience, we replace $\perp^{k_{\max}}$ with $\perp^{k_{\max}} \top$, which can be thought of fixing a dummy query at time $k_{\max} + 1$ which always outputs $+\infty$ regardless of inputs. In both cases, we can completely describe the output distribution the SVT with a positive random integer K. As a result, we will write $K \sim \mathcal{M}(D)$ and $K \sim \mathcal{M}(D')$ without loss of generality.

Also w.l.o.g., we assume thresholds T_i are all zero. There are two types of random variables in the algorithm: the threshold noise ρ and the query noise ν_i to each of the i queries, $\{\nu_i\}_{i=1}^{k+1}$. We will use $p_\rho(z)$ to denote the probability density of ρ, evaluated at z, and we will use $p(\nu_i)$ as the pdf of ν_i.

The probability of outputting o (or $K = k+1$), can be written explicitly as follows:

$$\Pr[\mathcal{M}(D') = o] = \int_{-\infty}^{+\infty} p_\rho(z)\Big(\prod_{i\leq k}\int_{-\infty}^{z-q_i(D')} p(\nu_i)d\nu_i\Big)\cdot\int_{z+q_{k+1}(D')}^{\infty} p(\nu_{k+1})d\nu_{k+1}dz.$$

Our goal of is to bound $\mathbb{E}_{o\sim\mathcal{M}(D')}\Big[\Big(\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\Big)^\alpha\Big]$ using the RDP functions of $\mathcal{M}_\rho$ and $\mathcal{M}_\nu$.

The key of the analysis relies on a sequence of fictitious queries $\tilde{q}_1, \tilde{q}_2, ...$ which mirrors the actual sequence of queries $q_1, q_2, ...$ that are adaptively selected. These fictitious queries satisfy for all $i = 1, 2, 3, ...$

$$\tilde{q}_i(x) = \begin{cases} q_i(D) + \triangle, & \text{when } x = D \\ q_i(D') & \text{otherwise} \end{cases} \tag{7.4}$$

The following lemma establishes that we can decompose the problem into one that involves the Renyi-divergence between a distribution induced by these fictitious queries and another distribution induced of the actual queries.

Lemma 7.6.1. *Consider Algorithm 11 with $c = 1$, i.e., the output sequence $o \in \{\perp^k \top | k = 0, 1, ..., \infty\}$, then we have*

$$\mathbb{E}_{o\sim\mathcal{M}(D')}\Big[\Big(\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\Big)^\alpha\Big] \leq \mathbb{E}_{z\sim p_\rho}\left[\Big(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\Big)^\alpha \underbrace{\mathbb{E}_{K\sim\mathcal{M}(D')}\left[\frac{\Big(\Pr[\mathcal{M}(D)=K|z,\tilde{\mathcal{Q}}]\Big)^\alpha}{(\Pr[\mathcal{M}(D')=K|z])^\alpha}\middle| z\right]}_{\textit{denoted by } (*)}\right] \tag{7.5}$$

where K is a random variable, denotes the number of $\perp$ plus 1 when the algorithm stops, and the explicit conditioning on $\tilde{\mathcal{Q}}$ indicates that the probability is evaluated by hypothetically running the algorithm on the fictitious queries $\tilde{q}_1, \tilde{q}_2, ... \in \tilde{\mathcal{Q}}$.

Proof: [Proof of Lemma 7.6.1] From the definition of Renyi DP, we have

$$\mathbb{E}_{o\sim D'}\left[\frac{\Pr[\mathcal{M}(D)=o]^{\alpha}}{\Pr[\mathcal{M}(D')=o]^{\alpha}}\right]=\sum_{k=0}^{\infty}\frac{\Pr[\mathcal{M}(D)=\perp^{k}\top]^{\alpha}}{\Pr[\mathcal{M}(D')=\perp^{k}\top]^{\alpha-1}} \tag{7.6}$$

Without loss of generality, we will replace o with k which measures the number of $\perp$s in o. By law of total expectation, we can condition on $\rho = z$

$$\begin{aligned}
&\Pr[\mathcal{M}(D)=k]=\mathbb{E}_{z\sim p_\rho}[\Pr[\mathcal{M}(D)=k|z]]\\
=&\mathbb{E}_{z\sim p_\rho}[\prod_{i\le k}\Pr[q_i(D)+\nu_i<z|z]\Pr[q_{k+1}(D)+\nu_i\ge z|z]]\\
=&\int_{-\infty}^{+\infty}p_\rho(z)\left(\prod_{i\le k}\int_{-\infty}^{z-q_i(D)}p(\nu_i)d\nu_i\right)\cdot\int_{z-q_{k+1}(D)}^{\infty}p(\nu_{k+1})d\nu_{k+1}dz\\
\overset{\overset{u:=z+\triangle}{\downarrow}}{=}&\int_{-\infty}^{+\infty}p_\rho(u-\triangle)\left(\prod_{i\le k}\int_{-\infty}^{u-\triangle-q_i(D)}p(\nu_i)d\nu_i\right)\cdot\int_{u-\triangle-q_{k+1}(D)}^{\infty}p(\nu_{k+1})d\nu_{k+1}du\\
=&\int_{-\infty}^{+\infty}p_\rho(u)\left(\frac{p_\rho(u-\triangle)}{p_\rho(u)}\right)\left(\prod_{i\le k}\int_{-\infty}^{u-\triangle-q_i(D)}p(\nu_i)d\nu_i\right)\cdot\int_{u-\triangle-q_{k+1}(D)}^{\infty}p(\nu_{k+1})d\nu_{k+1}du\\
=&\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)\left(\prod_{i\le k}\int_{-\infty}^{z-\triangle-q_i(D)}p(\nu_i)d\nu_i\right)\cdot\int_{z-\triangle-q_{k+1}(D)}^{\infty}p(\nu_{k+1})d\nu_{k+1}\right]
\end{aligned}$$

where in the last line, we rename the variable u back to z.

Substituting the above expression to the definition of RDP and apply Jensen's inequality

$$(7.6) = \sum_{k=0}^{\infty} \frac{\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)\left(\prod_{i\le k}\int_{-\infty}^{z-\triangle-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{z-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right]^{\alpha}}{\mathbb{E}_{z\sim p_\rho}\left[\left(\prod_{i\le k}\int_{-\infty}^{z-q_i(D')} p(\nu_i)d\nu_i\right)\cdot\int_{z--q_{k+1}(D')}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right]^{\alpha-1}}$$

$$\le \sum_{k=0}^{\infty} \mathbb{E}_{z\sim p_\rho} \frac{\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}(\prod_{i\le k}\int_{-\infty}^{z-\triangle-q_i(D)} p(\nu_i)d\nu_i)\int_{z-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right)^{\alpha}}{\left((\prod_{i\le k}\int_{-\infty}^{z-q_i(D')} p(\nu_i)d\nu_i)\int_{z-q_{k+1}(D')}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right)^{\alpha-1}} \tag{7.7}$$

The inequality applies Jensen's inequality to bivariate function $f(x, y) = x^\alpha y^{1-\alpha}$, which is jointly convex on $\mathcal{R}_+^2$ for $\alpha \in (1, +\infty)$.

Exchange the order of integral variable z and k in (7.7), we get (7.7) =

$$\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha} \sum_{k=0}^{\infty} \frac{\left((\prod_{i\le k}\int_{-\infty}^{z-\triangle-q_i(D)} p(\nu_i)d\nu_i)\int_{z-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right)^{\alpha}}{\left((\prod_{i\le k}\int_{-\infty}^{z-q_i(D')} p(\nu_i)d\nu_i)\int_{z-q_{k+1}(D')}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right)^{\alpha-1}}\right]$$

$$=\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha} \sum_{k=0}^{\infty} \frac{\left(\prod_{i=1}^{k}\Pr[q_i(D)+\triangle+\nu_i < z|z]\Pr[q_{k+1}(D)+\triangle+\nu_{k+1}\ge z|z]\right)^{\alpha}}{\left(\Pr_{\mathcal{M}(D')}[K=k+1|z]\right)^{\alpha-1}}\right] \tag{7.8}$$

$$=\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha} \mathbb{E}_{K\sim\mathcal{M}(D')}\left[\frac{\left(\prod_{i=1}^{K-1}\Pr[\tilde{q}_i(D)+\nu_i < z|z]\Pr[\tilde{q}_K(D)+\nu_K\ge z|z]\right)^{\alpha}}{(\Pr[\mathcal{M}(D')=K|z])^{\alpha}}\middle| z\right]\right]$$

$$=\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha} \mathbb{E}_{K\sim\mathcal{M}(D')}\left[\frac{\left(\Pr[\mathcal{M}(D)=K|z,\tilde{\mathcal{Q}}]\right)^{\alpha}}{(\Pr[\mathcal{M}(D')=K|z])^{\alpha}}\middle| z\right]\right] \tag{7.9}$$

which completes the proof.

To understand the last step: recall our definition of fictitious query $\tilde{q}_i$, which obeys $\tilde{q}_i(D) = q_i(D) + \triangle$ and $\tilde{q}_i(\tilde{D}) = q_i(\tilde{D})$ for all other dataset $\tilde{D} \neq D$. Observe that the expression in the numerator of (7.8) actually describes a valid probability distribution of K, which says the probability of $\mathcal{M}(D)$ stopping at time $K = k+1$ when the sequence of input is $\tilde{q}_1, ..., \tilde{q}_k + 1, ...$, i.e.,

$$\prod_{i=1}^{k} \Pr[q_i(D) + \Delta + \nu_i < z|z] \Pr[q_{k+1}(D) + \Delta + \nu_i \geq z|z] = \Pr_{\mathcal{M}(D)}[K = k+1|z, \tilde{q}_1, \tilde{q}_2, ...].$$

The conditioning on the sequence of queries might appear to be new, but recall that all our probabilities are conditioned on a sequence of queries that are chosen from $\mathcal{Q}(\triangle)$ or $\mathcal{Q}_+(\triangle)$ to begin with. They are just not written out explicitly. This is an instance, where we actually need to condition on a different set of queries to formally write down this valid probability distribution above. ■

Remark 7.6.2. The lemma de-convolves the moment of interests into the mixture of conditional moments of another two distributions that can be written down explicitly. The proof is delicate but informative, as it explicitly leveraging the fact that $\mathcal{M}_\nu$ is a *noise-adding* mechanism, so as to argue the implication of the randomization for a *different* query $\tilde{q}_i$ than the one that it seems to be intending for according to the algorithm q_i. There are several other novel components. We encourage readers to check it out in details.

A remarkable consequence of this lemma is that we can essentially cancel all factors concerning $\perp$s.

$$\begin{aligned}(*) =& \mathbb{E}_{K\sim\mathcal{M}(D')}\left[\left.\frac{\left(\prod_{i=1}^{K-1}\Pr[q_i(D) + \triangle + \nu_i < z|z]\Pr[q_K(D) + \triangle + \nu_i \geq z|z]\right)^\alpha}{\left(\prod_{i=1}^{K-1}\Pr[q_i(D') + \nu_i < z|z]\Pr[q_K(D') + \nu_i \geq z|z]\right)^\alpha}\right| z\right] \\ \leq& \mathbb{E}_{K\sim\mathcal{M}(D')}\left[\left.\frac{(\Pr[q_K(D) + \triangle + \nu_i \geq z|z])^\alpha}{(\Pr[q_K(D') + \nu_i \geq z|z])^\alpha}\right| z\right] \end{aligned} \tag{7.10}$$

The inequality in the last line uses the fact that q_i has a global sensitivity of $\triangle$, which implies that $\Pr[q_i(D) + \triangle + \nu_i < z|z] \leq \Pr[q_i(D') + \nu_i < z|z.$ for all i.

Further observe that $\tilde{q}_K$ has a sensitivity of $2\triangle$ since q_K has sensitivity $\triangle$. By the property of the noise-adding mechanism $\mathcal{M}_\nu$, it obeys $\epsilon_\nu(\alpha)$-RDP for all queries having having sensitivity $2\triangle$. Therefore, if $\epsilon_\nu(\infty) < +\infty$, then we can bound (7.10) with $e^{\alpha\epsilon_\nu(\infty)}$. In fact, this bound can be improved slightly if we directly work with (7.5), which we state as a lemma.

Lemma 7.6.3. *If $\epsilon_\nu(\infty) < +\infty$, then the expression $(*)$ in (7.5) obeys $(*) \leq e^{(\alpha-1)\epsilon_\nu(\infty)}$.*

Proof: We use a trick due to [Bun and Steinke, 2016] with some modifications.

First, check that by our trivial bounds

$$0 \leq \frac{\Pr[\mathcal{M}(D) = K|z, \tilde{\mathcal{Q}}]}{\Pr[\mathcal{M}(D') = K|z]} \leq e^\epsilon$$

Define random function $A(K)$ supported on $\{0, e^\epsilon\}$ such that $\mathbb{E}[A(K)|K] = \frac{\Pr[\mathcal{M}(D)=K|z,\tilde{\mathcal{Q}}]}{\Pr[\mathcal{M}(D')=K|z]}$.

Note that when $\alpha = 1$

$$\Pr[A(K) = e^\epsilon] \cdot e^\epsilon = \mathbb{E}_K[\mathbb{E}[A(K)|K]] = \mathbb{E}_{K\sim\mathcal{M}(D')}\left[\frac{\Pr[\mathcal{M}(D) = K|z, \tilde{\mathcal{Q}}]}{\Pr[\mathcal{M}(D') = K|z]}\right] = 1.$$

The first moment is equal to 1 critically relies on our construction where the numerator in the expectation of $(*)$ is the αth power of a valid probability distributions.

This implies that $\Pr[A(K) = e^\epsilon] = e^{-\epsilon}$, therefore

$$(*) = \mathbb{E}_K[\mathbb{E}[A(K)|K]^\alpha] \overset{\substack{\text{Jensen}\\ \downarrow}}{\leq} \mathbb{E}[\mathbb{E}[A(K)^\alpha|K]] = \mathbb{E}[A(K)^\alpha] = \Pr[A(K) = e^\epsilon] \cdot e^{\alpha\epsilon} = e^{(\alpha-1)\epsilon},$$

which completes the proof. ■

Now we are ready to prove the three claims of Theorem 7.2.1.

The claim (7.3)**:** Substitute the the above bound into Lemma 7.6.1, we get:

$$\begin{aligned}&\mathbb{E}_{o\sim\mathcal{M}(D')}\left[\left(\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(D')=o]}\right)^{\alpha}\right]\\ \overset{\substack{\text{Lemma 7.6.1 and 7.6.3}\\ \downarrow}}{\le}&\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha}\right]e^{(\alpha-1)\epsilon_\nu(\infty)}\le e^{(\alpha-1)\epsilon_\rho(\alpha)}e^{(\alpha-1)\epsilon_\nu(\infty)}.\end{aligned}$$

where the second inequality in the last line uses the definition of RDP of $\mathcal{M}_\rho$, for a trivial query $q(D)=-\Delta, q(D')=0$. (7.3) follows by simply taking $\log(\cdot)/(\alpha-1)$ on both sides.

The claim (7.1) **and** (7.2)**:** To get the other two bounds, we need an alternative analysis of $(**)$. To avoid crowded notations, we drop the conditioning on z from $\Pr[\cdot|\rho = z]$. By the definition of expectation,

$$\begin{aligned}(*)\le(7.10)=\;&\sum_{k=0}^{\infty}\prod_{i=0}^{k}\Pr[q_i(D')+\nu_i<z]\Pr[q_{k+1}(D')+\nu_{k+1}\ge z]\frac{\Pr[\tilde{q}_{k+1}(D)+\nu_{k+1}\ge z]^{\alpha}}{\Pr[\tilde{q}_{k+1}(D')+\nu_{k+1}\ge z]^{\alpha}}\\ \overset{\substack{\tilde{q}=q\text{ on }D'\\ \downarrow}}{=}\;&\sum_{k=0}^{\infty}\prod_{i=0}^{k}\Pr[q_i(D')+\nu_i<z]\frac{\Pr[\tilde{q}_{k+1}(D)+\nu_{k+1}\ge z]^{\alpha}}{\Pr[\tilde{q}_{k+1}(D')+\nu_{k+1}\ge z]^{\alpha-1}}\\ \overset{\substack{\text{Lemma ??}\\ \downarrow}}{\le}\;&\left(\sum_{k=0}^{\infty}\prod_{i=0}^{k}\Pr[\tilde{q}_i(D')+\nu_i<z]\right)\cdot e^{\epsilon(\alpha)(\alpha-1)}. \qquad (7.11)\end{aligned}$$

In the last line, we applied the "indistinguishability" property of an RDP mechanism in Lemma **??** for the particular event $S = x\in\mathbb{R}|x\ge z$, for the random-variable $\mathcal{M}(D,\tilde{q}_{k+1})$ and $\mathcal{M}(D',\tilde{q}_{k+1})$ in the numerator and denominator respectively.

The issue is how to proceed. $\sum_{k=0}^{\infty}\prod_{i=0}^{k}\Pr[\tilde{q}_i(D')+\nu_i<z]$ does not sum to 1 because $\prod_{i=0}^{k}\Pr[\tilde{q}_i(D')+\nu_i<z]$ is not a probability distribution of k. The saving grace is the

following alternative definition of expectation.

Lemma 7.6.4. *For a non-negative random variable X, $\mathbb{E}[X] = \int_0^\infty \Pr[X > x]dx$.*

Recall that K is the first index of $\top$, we can rewrite $\prod_{i=0}^{k} \Pr[\tilde{q}_i(D') + \nu_i < z]$ as $\Pr_{D'}[K > k|z]$. Thus

$$\sum_{k=0}^{\infty} \prod_{i=0}^{k} \Pr[\tilde{q}_i(D') + \nu_i < z] = \sum_{k=0}^{\infty} \Pr_{D'}[K > k|z] = \mathbb{E}[K|z] \tag{7.12}$$

It follows that

$$\mathbb{E}_{o\sim\mathcal{M}(D')}\left[\left(\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}\right)^{\alpha}\right] \leq \mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\alpha}\mathbb{E}[K|z]\right]e^{\epsilon_\nu(\alpha)(\alpha-1)}$$

Claim (7.1) uses $\mathbb{E}[K|z] \leq k_{\max} + 1$. By using a different Holder's inequality with conjugate pair γ and $\gamma* := \gamma/(\gamma - 1)$, we obtain

$$\mathbb{E}_{o\sim\mathcal{M}(D')}\left[\left(\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}\right)^{\alpha}\right] \leq \mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)^{\gamma^*\alpha}\right]^{1/\gamma^*} \cdot \left(\mathbb{E}_{z\sim p_\rho}\left[\mathbb{E}[K|z]^{\gamma}\right]\right)^{1/\gamma} \cdot e^{\epsilon_\nu(\alpha)(\alpha-1)}$$

(7.2) follows by taking $\log(\cdot)/(\alpha - 1)$ on both sides and applying the definition of RDP. This completes the proof of Theorem 7.2.1.

Proposition 7.6.5 (Restatement of Proposition 7.2.5 with mroe details)**.** *Let Algorithm 11 be instantiated with $\mathcal{Q}_+(\triangle)$, $\mathcal{M}_\rho$ and $\mathcal{M}_\nu$ be Gaussian mechanism with parameter σ_1 and σ_2. Then for all $T < +\infty$ and $\gamma > 1$ such that $\sigma_2 > \sqrt{\gamma}\sigma_1$, Algorithm 11 with $c = 1$ halts with K rounds satisfying*

$$\mathbb{E}_\rho[\mathbb{E}[K|\rho = z]^\gamma] \leq \int_{-\infty}^{\infty} \frac{1}{\sigma_1}\phi(z/\sigma_1)\left(\frac{\Phi((T+z)/\sigma_2)}{1 - \Phi((T+z)/\sigma_2)}\right)^{\gamma} dz < +\infty, \tag{7.13}$$

where $\phi(x) = \frac{e^{-x^2}}{\sqrt{2\pi}}$ and $\Phi(x) = \int_{-\infty}^{x} \phi(x)dx$ are the pdf and CDF of the standard normal *distribution. If $\sigma_2 \geq \sqrt{\gamma+1}\sigma_1$, then a more interpretable bound of the above is*

$$\mathbb{E}[\mathbb{E}[K|\rho = z]^\gamma] \leq 1 + (c_\gamma \sqrt{2\pi} \max\{\frac{T(1+\gamma)}{\sigma_1}, 1\})^\gamma (1+\gamma)^{1/2} e^{\frac{\gamma T^2}{2\sigma_1^2}}$$

where c_γ is a universal constant that comes from the moments bounds and depends only on γ. For the special case when $\gamma = 2$, and $\sigma_2 = \sqrt{3}\sigma_1$, we get $\mathbb{E}[\mathbb{E}[K|\rho = z]^2] \leq 1 + 2\sqrt{3}\pi(1 + \frac{9T^2}{\sigma_1^2})e^{\frac{T^2}{\sigma_1^2}}$.

Proof: [Proof of Proposition 7.6.5] Consider the case when all queries are non-negative, and the threshold T is given, then the datasets that maximizes all moments of $K|\rho = z$ for all z are given by $f_i(D) = 0$ for all i. Notice that $K|\rho = z$ follows a Negative Binomial Distribution, thus

$$\mathbb{E}[K|z] = \frac{F_v[T+z]}{1 - F_v[T+z]},$$

where F_v is the cumulative density function (CDF) of the noise v. The moments of $\mathbb{E}[K|z]$, when exists, can be computed by numerical integration. When $z \sim \mathcal{N}(0, \sigma_1^2)$ and $v \sim \mathcal{N}(0, \sigma_2^2)$ for $\sigma_2 > \sigma_1\sqrt{\gamma}$, we can work out bounds of the γth moments of $\mathbb{E}[K|z]$.

Let ϕ be the standard normal density function and Φ be the CDF. There is a lower bound of the Gaussian tail for all $x > 0$

$$1 - \Phi(x) \geq \frac{x}{x^2+1}\phi(x)$$

Thus for $y \geq -T + \sigma_2$, we have

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}[K|z]^\gamma\right] = \int_{-\infty}^{\infty} \frac{1}{\sigma_1}\phi(z/\sigma_1)\left(\frac{\Phi((T+z)/\sigma_2)}{1-\Phi((T+z)/\sigma_2)}\right)^\gamma dz \\
&= \int_{-\infty}^{y} \frac{1}{\sigma_1}\phi(z/\sigma_1)\left(\frac{\Phi((T+z)/\sigma_2)}{1-\Phi((T+z)/\sigma_2)}\right)^\gamma dz + \int_{y}^{\infty} \frac{1}{\sigma_1}\phi(z/\sigma_1)\left(\frac{\Phi((T+z)/\sigma_2)}{1-\Phi((T+z)/\sigma_2)}\right)^\gamma dz \\
&\leq \int_{-\infty}^{y} \frac{1}{\sigma_1}\phi(z/\sigma_1)dz + \int_{y}^{\infty} \frac{1}{\sigma_1}\phi(z/\sigma_1)\left(\frac{\frac{(T+z)^2}{\sigma_2^2}+1}{\frac{(T+z)}{\sigma_2}\phi((T+z)/\sigma_2)}\right)^\gamma dz \\
&\overset{\overset{T+y\geq\sigma_2}{\downarrow}}{\leq} \Phi(y/\sigma_1) + \int_{y}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{z^2}{2\sigma_1^2}} (2\pi)^{\gamma/2} e^{\frac{\gamma(T+z)^2}{2\sigma_2^2}} 2^\gamma \frac{(T+z)^\gamma}{\sigma_2^\gamma} dz \\
&\overset{\overset{u:=z+T,\ \tilde{\sigma}:=(\frac{1}{\sigma_1^2}-\frac{\gamma}{\sigma_2^2})^{-1/2}}{\downarrow}}{=} \Phi(y/\sigma_1) + (2\pi)^{\frac{\gamma}{2}}\frac{\tilde{\sigma}}{\sigma_1}\int_{T+y}^{\infty} \frac{u^\gamma}{\sigma_2^\gamma}\frac{1}{\sqrt{2\pi}\tilde{\sigma}} e^{-\frac{u^2}{2\tilde{\sigma}^2}+\frac{2uT}{2\sigma_1^2}-\frac{T^2}{2\sigma_1^2}} du \\
&= \Phi(y/\sigma_1) + (2\pi)^{\frac{\gamma}{2}}\frac{\tilde{\sigma}}{\sigma_1} e^{-\frac{T^2}{2\sigma_1^2}+\frac{T^2\tilde{\sigma}^2}{2\sigma_1^4}} \int_{T+y}^{\infty} \frac{u^\gamma}{\sigma_2^\gamma}\frac{1}{\sqrt{2\pi}\tilde{\sigma}} e^{-\frac{(u-T\frac{\tilde{\sigma}^2}{\sigma_1^2})^2}{2\tilde{\sigma}^2}} du \\
&\overset{\overset{\text{take } |\cdot| \text{ and relax integral}}{\downarrow}}{\leq} \Phi(y/\sigma_1) + (2\pi)^{\frac{\gamma}{2}}\frac{\tilde{\sigma}}{\sigma_1} e^{\frac{\gamma T^2}{2(\sigma_2^2-\gamma\sigma_1^2)}} \mathbb{E}_{X\sim\mathcal{N}(\frac{T\tilde{\sigma}}{\sigma_1^2},1)}[|X|^\gamma]
\end{aligned}$$

where $\mathbb{E}_{X\sim\mathcal{N}(\frac{T\tilde{\sigma}}{\sigma_1^2},1)}[|X|^\gamma]$ is the mth non-central moments which is on the order of $\max\{\frac{T\tilde{\sigma}}{\sigma_1^2}, 1\}^\gamma$ — and can be evaluated in a closed-form. Finally, we can simply take $y\ -T + \sigma_2$.

Now, suppose we take $\sigma_2 = \sqrt{1+\gamma}\sigma_1$, then we get $\tilde{\sigma} = \sigma_2$. The above bound simplifies to:

$$\mathbb{E}[\mathbb{E}[K|z]^\gamma] \leq 1 + (c_\gamma\sqrt{2\pi}\max\{\frac{T(1+\gamma)}{\sigma_1}, 1\})^\gamma(1+\gamma)^{1/2}e^{\frac{\gamma T^2}{2\sigma_1^2}}$$

where c_γ is a universal constant that comes from the moments bounds and depends only on γ. If $\gamma = 2$ and $\sigma_2 = \sqrt{3}\sigma_1$, then $\tilde{\sigma} = \sigma_2$, and

$$\mathbb{E}[\mathbb{E}[K|z]^2] \leq 1 + 2\sqrt{3}\pi(1+\frac{9T^2}{\sigma_1^2})e^{\frac{T^2}{\sigma_1^2}}.$$

■

7.6.2 RDP analysis with $c \geq 1$, proof of Theorem 7.2.7

Theorem 7.6.6 (Restatement of Theorem 7.2.7, RDP for length-capped SVT with $c > 1$)**.** *The generalized SVT with cut-off parameter $c > 1$ and a maximum length is $k_{\max}$ obeys that*

$$\mathbb{D}_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \epsilon_\rho(\alpha) + c\epsilon_\nu(\alpha) + \frac{\log \sum_{k=0}^{c} \binom{k_{\max}}{k}}{\alpha - 1}.$$

*After a careful revision, we found there is a minor typo in the statement of Theorem 7.2.7 (the $1/(\alpha-1)$ term shouldn't be there) we provide the correct version above.

The proof follows a similar sequence of arguments to that we presented for $c = 1$.

When $c > 1$, the output space of the algorithm is $S = \{\top, \bot\}^\ell, \ell = 0, 1, ..., k_{\max}$with the additional restriction that the number of $\top$s are smaller than c. Denote $I_\bot := \{i : o_i =\bot\}$ and $I_\top := \{j : o_j = \top\}$. Then we can write the probability of outputting o as following:

$$\Pr[\mathcal{M}(D) = o] = \int_{-\infty}^{+\infty} p_\rho(z)\Big(\prod_{i\in I_\bot}\int_{-\infty}^{z-q_i(D)} p(\nu_i)d\nu_i\Big)\Big(\prod_{j\in I_\top}\int_{z+q_j(D)}^{\infty} p(\nu_j)d\nu_j\Big)dz$$

Similarly, we have

$$\mathbb{E}_o\Big[\Big(\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}\Big)^\alpha\Big] = \sum_{o\in S}\Pr[\mathcal{M}(D') = o]\Big(\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}\Big)^\alpha \qquad (*)$$

Apply the same logic from the proof for $c = 1$, we can upper bound $\Pr[\mathcal{M}(D) = o]$ in the following

$$\Pr[\mathcal{M}(D)=o] \leq \mathbb{E}_{z\sim p_\rho}\bigg(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\prod_{i\in I_\perp}\int_{-\infty}^{z-q_i(D')}p(\nu_i)d\nu_i\bigg)\bigg(\prod_{j\in I_\top}\int_{z+\triangle+q_j(D)}^{\infty}p(\nu_j)d\nu_j\bigg) \tag{7.14}$$

Then apply Jensen' inequality to $\frac{\Pr[\mathcal{M}(D)=o]^\alpha}{\Pr[\mathcal{M}(D')=o]^{\alpha-1}}$, we have

$$(7.14) \leq \sum_{o\in S}\mathbb{E}_{z\sim p_\rho}\frac{\bigg(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\prod_{i\in I_\perp}\int_{-\infty}^{z-q_i(D')}p(\nu_i)d\nu_i\bigg)^\alpha\bigg(\prod_{j\in I_\top}\int_{z+\triangle+q_j(D)}^{\infty}p(\nu_j)d\nu_j\bigg)^\alpha}{\bigg(\prod_{i\in I_\perp}\int_{-\infty}^{z-q_i(D')}p(\nu_i)d\nu_i\bigg)^{\alpha-1}\bigg(\prod_{j\in I_\top}\int_{z+q_j(D')}^{\infty}p(\nu_j)d\nu_j\bigg)^{\alpha-1}} \tag{7.15}$$

Exchange the order of integral in z and o, we get

$$(7.15) = \underbrace{\mathbb{E}_{z\sim p_\rho}\bigg(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\bigg)^\alpha\sum_{o\in S}\Pr[\prod_{i\in I_\perp}q_i(D')+\nu_i<z]}_{\text{denote by}(**)}\cdot\frac{\bigg(\prod_{j\in I_\top}\Pr[q_j(D)+\nu_j+\triangle\geq z]\bigg)^\alpha}{\bigg(\prod_{j\in I_\top}\Pr[q_j(D')+\nu_j\geq z]\bigg)^{\alpha-1}}$$

In the case of length-capped SVT, the algorithm stops whenever $|o|\geq k_{\max}$ or $|I_\top|\geq c$. By the fact that probabilities ≤ 1, we use the following crude bound

$$\sum_{o\in S}\Pr[\prod_{i\in I_\perp}q_i(D')+\nu_i<z]\leq\sum_{o\in S}1=|S|,$$

i.e., the cardinality of the output space, which is bounded from above by $\sum_{k=0}^{c}\binom{k_{\max}}{k}$.

Moreover, we can bound the $\mathbb{E}_{z\sim p_\rho}\bigg(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\bigg)^\alpha$ term with $e^{(\alpha-1)\epsilon_\rho(\alpha)}$ using the definition of RDP (with a trivial query that outputs 0 and $\triangle$ for D and D' as we constructed before). Therefore, $(**)$ is bounded by $e^{(\alpha-1)\epsilon_\rho(\alpha)}\cdot\sum_{k=0}^{c}\binom{k_{\max}}{k}$.

For the second part $\frac{(\prod_{j\in I_\top}\Pr[q_j(D)+\nu_j+\triangle\geq z])^\alpha}{(\prod_{j\in I_\top}\Pr[q_j(D')+\nu_j\geq z])^{\alpha-1}}$, we apply the same trick of defining a sequence of fictitious queries $\tilde{q}_1, ..., \tilde{q}_{k_{\max}}$ as in 7.4. For each $j \in I_\top$, $\frac{\Pr[\tilde{q}_j(D)+\nu_j+\triangle \geq z]}{\Pr[\tilde{q}_j(D')+\nu_j \geq z])^{\alpha-1}} \leq e^{\epsilon_\nu(\alpha)(\alpha-1)}$ using the "indistinguishability" property of an RDP mechanism. Since $|I_\top| \leq c$, the second part is bounded by $e^{c\epsilon_\nu(\alpha)(\alpha-1)}$.

Bibliography

Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS-16)*, pages 308–318. ACM, 2016.

Apple, Differential Privacy Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 2017.

Shahab Asoodeh, Jiachun Liao, Flavio P Calmon, Oliver Kosut, and Lalitha Sankar. Three variants of differential privacy: Lossless conversion and applications. *IEEE Journal on Selected Areas in Information Theory*, 2(1):208–222, 2021.

Borja Balle and Yu-Xiang Wang. Improving gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. *International Conference in Machine Learning (ICML)*, 2018.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Preprint*, 2018.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2496–2506. PMLR, 2020.

Gilles Barthe and Federico Olmedo. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In *International Colloquium on Automata, Languages, and Programming*, pages 49–60. Springer, 2013.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS-14)*, pages 464–473. IEEE, 2014.

Raef Bassily, Om Thakkar, and Abhradeep Guha Thakurta. Model-agnostic private learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *Conference on Innovations in Theoretical Computer Science (ITCS-13)*, pages 97–110. ACM, 2013.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning*, pages 1006–1014, 2015.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

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

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 634–649. IEEE, 2015.

Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1306–1325. SIAM, 2017.

Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *STOC-18*, 2018.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *The Journal of Machine Learning Research*, 12:1069–1109, 2011.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Yuval Dagan and Vitaly Feldman. Pac learning with stable and private predictions. In *Conference on Learning Theory*, pages 1389–1410. PMLR, 2020.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Chris Decarolis, Mukul Ram, Seyed Esmaeili, Yu-Xiang Wang, and Furong Huang. An end-to-end differentially private latent dirichlet allocation using a spectral algorithm. In *International Conference on Machine Learning*, pages 2421–2431. PMLR, 2020.

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin IP Rubinstein. Robust and private bayesian inference. In *Algorithmic Learning Theory*, pages 291–305. Springer, 2014.

Zeyu Ding, Yuxin Wang, Danfeng Zhang, and Daniel Kifer. Free gap information from the differentially private sparse vector and noisy max mechanisms. *arXiv preprint arXiv:1904.12773*, 2019.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society, Series B*, 2021. to appear.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Cynthia Dwork. Differential privacy. In *International conference on Automata, Languages and Programming*, pages 1–12. Springer-Verlag, 2006.

Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *Conference On Learning Theory*, pages 1693–1702. PMLR, 2018.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):211–407, 2013.

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

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

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390. ACM, 2009.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 51–60. IEEE, 2010.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. *arXiv preprint arXiv:1411.2664*, 2014a.

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

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *ACM symposium on Theory of computing (STOC-14)*, pages 11–20. ACM, 2014c.

Charles L Epstein and John Schotland. The bad truth about laplace's transform. *SIAM review*, 50(3):504–520, 2008.

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*, pages 1054–1067. ACM, 2014.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems*, 34:28080–28091, 2021.

James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving bayesian data analysis. In *Conference on Uncertainty in Artificial Intelligence (UAI-16)*, pages 192–201. AUAI Press, 2016.

Marco Gaboardi, James Honaker, Gary King, Kobbi Nissim, Jonathan Ullman, Salil Vadhan, and Jack Murtagh. Psi (ψ): a private data sharing interface. In *Theory and Practice of Differential Privacy*, New York, NY, 2016 2016. URL `https://arxiv.org/abs/1609.04340`.

Quan Geng and Pramod Viswanath. The optimal mechanism in differential privacy. In *2014 IEEE international symposium on information theory*, pages 2371–2375. IEEE, 2014.

Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. Renyi differential privacy mechanisms for posterior sampling. In *Advances in Neural Information Processing Systems*, pages 5295–5304, 2017.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *arXiv preprint arXiv:2106.02848*, 2021.

Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. *arXiv preprint arXiv:2203.00263*, 2022.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70. IEEE, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International Conference on Machine Learning (ICML-15)*, 2015.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3): 793–826, 2011.

Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, pages 457–476. Springer, 2013.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1:41, 2012.

Antti Koskela and Antti Honkela. Computing differential privacy guarantees for heterogeneous compositions using fft. *arXiv preprint arXiv:2102.12412*, 2021.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 2560–2569. PMLR, 2020a.

Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight approximate differential privacy for discrete-valued mechanisms using fft. *arXiv preprint arXiv:2006.07134*, 2020b.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Tech Report*, 2009.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195, 2015.

Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 32–33. ACM, 2012.

Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

Xiyang Liu, Weihao Kong, and Sewoong Oh. Differential privacy and robust statistics in high dimensions. *arXiv preprint arXiv:2111.06578*, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

Min Lyu, Dong Su, and Ninghui Li. Understanding the sparse vector technique for differential privacy. *Proceedings of the VLDB Endowment*, 10(6):637–648, 2017.

Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR-18)*, 2018.

Sebastian Meiser and Esfandiar Mohammadi. Tight on budget? tight bounds for r-fold approximate differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS-18)*, pages 247–264, 2018.

Kentaro Minami, HItomi Arai, Issei Sato, and Hiroshi Nakagawa. Differential privacy without sensitivity. In *Advances in Neural Information Processing Systems*, pages 956–964, 2016.

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

Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Jack Murtagh and Salil Vadhan. The complexity of computing the optimal composition of differential privacy. In *Theory of Cryptography Conference*, pages 157–175. Springer, 2016.

Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*, 2011.

Frank Nielsen and Richard Nock. On the chi square and higher-order chi distances for approximating f-divergences. *IEEE Signal Processing Letters*, 21(1):10–13, 2014.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *ACM symposium on Theory of computing (STOC-07)*, pages 75–84. ACM, 2007.

N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.

Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*, 2021.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations (ICLR-18)*, 2018.

Mijung Park, James Foulds, Kamalika Chaudhuri, and Max Welling. Variational bayes in private settings (vips). *arXiv preprint arXiv:1611.00340*, 2016.

Rachel Redberg and Yu-Xiang Wang. Privately publishable per-instance privacy. *Advances in Neural Information Processing Systems*, 34, 2021.

Rachel Redberg, Yuqing Zhu, and Yu-Xiang Wang. Generalized ptr: User-friendly recipes for data-adaptive algorithms with differential privacy. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 3977–4005. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/redberg23a.html`.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

Ian M Rodriguez, William N Sexton12, Phyllis E Singer, and Lars Vilhuber. The modernization of statistical disclosure limitation at the us census bureau.

Ryan M Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. *Advances in Neural Information Processing Systems*, 29, 2016.

Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774, 2010.

Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model assisted survey sampling*. Springer Science & Business Media, 2003.

Adam Smith. Information, privacy and stability in adaptive data analysis. *arXiv preprint arXiv:1706.00820*, 2017.

Adam Smith and Aaron Roth. Lecture notes in algorithmic foundation of adaptive data analysis, Nov 2017. URL `https://adaptivedataanalysis.com/lecture-schedule-and-notes/`.

David M Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *Proceedings on privacy enhancing technologies*, 2019(2):245–269, 2019.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Conference on Signal and Information Processing*, 2013.

Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and David Megías. Individual differential privacy: A utility-preserving formulation of differential privacy guarantees. *IEEE Transactions on Information Forensics and Security*, 12(6):1418–1429, 2017.

Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850. PMLR, 2013.

Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YTWGvpFOQD-`.

Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

Igor Vajda. χ^α-divergence and generalized fisher information. In *Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, page 223. Academia, 1973.

Laurens van der Maaten and Awni Hannun. The trade-offs of private prediction. *arXiv preprint arXiv:2007.05089*, 2020.

Jiachen T Wang, Saeed Mahloujifar, Shouda Wang, Ruoxi Jia, and Prateek Mittal. Renyi differential privacy of propose-test-release and applications to private and robust machine learning. *arXiv preprint arXiv:2209.07716*, 2022.

Yu-Xiang Wang. Per-instance differential privacy. *Journal of Privacy and Confidentiality, to appear.*, 2018a.

Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In *Uncertainty in Artificial Intelligence (UAI-18)*, 2018b.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning (ICML-15)*, pages 2493–2502, 2015.

Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *Journal of Machine Learning Research*, 17(183):1–40, 2016.

Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics (AISTATS-19)*, 2019a.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019b.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

Xuandong Zhao, Zhiguo Yu, Ming Wu, and Lei Li. Compressing sentence representation for semantic retrieval via homomorphic projective distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 774–781, May 2022.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

Yuqing Zhu and Yu-Xiang Wang. Poisson subsampled rényi differential privacy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7634–7642. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/zhu19c.html`.

Yuqing Zhu and Yu-Xiang Wang. Improving sparse vector technique with renyi differential privacy. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20249–20258. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/e9bf14a419d77534105016f5ec122d62-Paper.pdf`.

Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4782–4817. PMLR, 28–30 Mar 2022. URL `https://proceedings.mlr.press/v151/zhu22c.html`.

Yuqing Zhu, Xuandong Zhao, Chuan Guo, and Yu-Xiang Wang. " private prediction strikes back!"private kernelized nearest neighbors with individual renyi filter. *arXiv preprint arXiv:2306.07381*, 2023.