Communication Efficient and Differentially Private Optimization

Shuli Jiang

Thesis Proposal

November 18, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Gauri Joshi, *Chair*
Steven Wu
Zachary Manchester
Swanand Kadhe, *IBM Research*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

Abstract

In recent years, the integration of communication efficiency and differential privacy in distributed optimization has gained significant attention, motivated by large-scale applications such as Federated Learning (FL), where both data privacy and efficient communication are critical. This thesis explores the development of novel techniques to address these challenges, with a focus on distributed mean estimation, differentially private prediction, and private optimization for empirical risk minimization.

The first part of this work addresses communication-efficient distributed vector mean estimation, an essential subroutine in distributed optimization and FL. We propose the Rand-Proj-Spatial family estimator which utilizes cross-client correlation to reduce the estimation error under fixed communication cost, by projecting client vectors into a random subspace using a Subsampled Randomized Hadamard Transform (SRHT). This approach captures cross-client correlation more effectively, demonstrating substantial performance gains over conventional sparsification techniques in various distributed optimization tasks.

The second part of this work focuses on maximizing the privacy-utility trade-offs in differentially private prediction through majority ensembling. We introduce the Data-dependent Randomized Response Majority (DaRRM) framework, which generalizes all private majority ensembling algorithms through a data-dependent noise function. Based on DaRRM, we propose a computationally tractable optimization procedure for maximizing utility under a fixed privacy loss. Empirical results demonstrate DaRRM's effectiveness in private label ensembling for image classification, showing significant utility improvements over existing baselines.

The third part of this work investigates differentially private optimization in solving empirical risk minimization using shuffled gradient methods. Unlike conventional private optimizers such as DP-SGD, which benefits from privacy amplification by subsampling, shuffled gradient methods face unique challenges in privacy and convergence. We develop a theoretical framework for analyzing Incremental Gradient (IG) methods, the most basic form of shuffled gradient methods, that enables noise injection for privacy and the use of surrogate objectives, introducing a new dissimilarity metric to measure the difference between true and surrogate objectives. Leveraging privacy amplification by iteration, we establish the first empirical excess risk bound for differentially private IG (DP-IG),

and show how interleaving public data in training can further improve privacy-convergence trade-offs in DP-IG.

Finally, we introduce two proposed works along the line of differentially private optimization. First, we aim to extend our theoretical framework to analyze Shuffle Once (SO) and Random Reshuffling (RR), two practical shuffled gradient methods beyond Incremental Gradient (IG) methods. This will enable us to understand their private counterparts, DP-SO and DP-RR, where privacy analysis is more complex due to a lack of understanding on privacy amplification through shuffling. Second, we plan to extend our framework from a local to a distributed or decentralized setting to analyze convergence rates of distributed shuffled gradient methods in both private and non-private contexts, while also investigating the impact of data heterogeneity among clients on convergence in this setting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Challenges in Distributed Learning

It is common to see massive amounts of data being generated and collected on edge client devices such as smartphones, tablets, and IoT sensors. The diverse data from different edge clients are extremely useful to train machine learning models [74], such as a model that enables next-word prediction, that can be used by those clients. However, the traditional approach of transferring data directly from clients to a central server for model training poses significant challenges. Edge clients often have limited communication bandwidth and limited computation power, making the transfer of large datasets expensive and slow. Moreover, the sensitive nature of client data raises privacy concerns, as laws and regulations frequently prohibit direct sharing of such data with centralized servers.

To address these issues, distributed learning paradigms have emerged as a promising solution. A notable example is federated learning [57], which enables collaborative model training without requiring raw data to leave individual devices. In a distributed learning setup, as illustrated by Figure 1.1, a central server coordinates with multiple edge clients. Each client performs local training on its own device using its own data, and the central server aggregates these locally trained models to produce a global model. This distributed learning approach helps alleviate the privacy risks and communication costs associated with a direct data transfer.

Despite its advantages, the distributed learning paradigm introduces two key

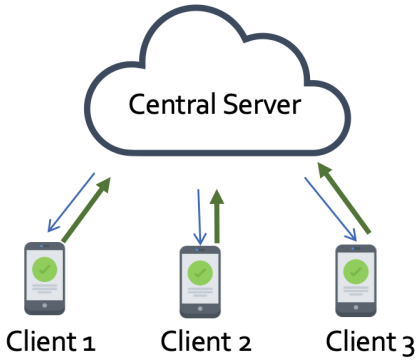

Figure 1.1: A distributed learning setup.

challenges. The first challenge is communication efficiency, e.g., [99, 101]. Edge clients often operate under bandwidth constraints, which makes transferring large machine learning models to and from the server infeasible. With the rapid growth in the size of modern machine learning models, the need to minimize the communication overhead between clients and the server becomes critical. Strategies to reduce the number of bits sent during training are essential for enabling efficient distributed learning.

The second challenge is ensuring data privacy. While federated learning keeps data on clients' devices, it does not completely eliminate the risk of information leakage. Research in security has demonstrated that it is possible to infer sensitive training data from the model parameters shared with the server [41]. Addressing this issue requires more rigorous privacy-preserving techniques. Differential privacy [32], a mathematically rigorous framework for quantifying privacy risks, has become the standard for reducing the risks of exposing sensitive information in distributed learning systems [115]. Algorithms typically achieve privacy guarantees by injecting noise, which inevitably impacts utility. The greater the privacy requirements, the more noise is added, reducing the algorithm's usefulness in practical applications. Hence, there is always a privacy-utility trade-off. Consequently, a key objective in designing private algorithms is to maximize this trade-off.

This thesis aims to develop novel techniques to address the two challenges in distributed learning, with a particular focus on improving communication efficiency and maximizing the privacy-utility trade-offs of differentially private algorithms in subroutines useful in distributed learning.

1.2 Roadmap

Chapter 2 starts by focusing on communication efficiency, where we study the problem of distribtued vector mean estimation under communication constraints. This problem is often used as a subroutine in distributed optimization and Federated Learning. We propose a new algorithm that can more effectively leverage cross-client correlation, a practically available side information, to improve the estimation accuracy while fixing the communication budget.

Chapter 3 shifts the focus to differential privacy, where we study the problem of private majority ensembling, which has applications in distributed private prediction and semi-supervised knowledge transfer.

Chapter 4 combines differential privacy with optimization. We study the private version of a specific type of optimization algorithms, Incremental Gradient (IG) method, which belongs to the family of shuffled gradient methods, for solving empirical risk minimization (ERM) problems. Unlike stochastic gradient descent (SGD) and its private counterpart, DP-SGD, private shuffled gradient methods remain underexplored. We further show how to leverage public data samples to maximize the privacy-convergence trade-offs in private IG.

Lastly, in Chapter 5, we propose two directions to work on along the line of private optimization. First, we propose an extension based on current results to study the private versions of Shuffle Once (SO) and Random Reshuffling (RR), two more practical shuffled gradient methods, beyond IG. Second, we propose to explore distributed and decentralized shuffled gradient methods, focusing on the interplay of differential privacy, communication efficiency, and client data heterogeneity on convergence.

Chapter 2

Communication Efficiency: Correlated Distributed Mean Estimation

This chapter is based on the following work:
Shuli Jiang, Pranay Sharma, Gauri Joshi. "Correlation Aware Sparsified Mean Estimation Using Random Projection". The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.

As previously mentioned, edge clients involved in the distributed training of a machine learning model often face constraints in communication bandwidth. Consequently, in scenarios where the global model to be trained is large, it becomes essential for each client to minimize the number of bits sent to the server. This chapter focuses on addressing the communication efficiency challenges in distributed learning.

We study the problem of communication-efficient distributed vector mean estimation[1], a commonly used subroutine in distributed optimization and Federated Learning (FL). Rand-k sparsification is a commonly used technique to reduce communication cost, where each client sends $k < d$ of its coordinates to the server. However, Rand-k is agnostic to any correlations, that might exist between clients in practical scenarios. The recently proposed Rand-k-Spatial estimator leverages the cross-client correlation information at the server to improve Rand-k's performance. Yet, the

[1]Note we focus on the empirical mean estimation, where the vectors at the clients are fixed. This is as opposed to statistical mean estimation, where the vectors are drawn from some distribution.

performance of Rand-k-Spatial is suboptimal. We propose the Rand-Proj-Spatial estimator with a more flexible encoding-decoding procedure, which generalizes the encoding of Rand-k by projecting the client vectors to a random k-dimensional subspace. We utilize Subsampled Randomized Hadamard Transform (SRHT) as the projection matrix and show that Rand-Proj-Spatial with SRHT outperforms Rand-k-Spatial, using the correlation information more efficiently. Furthermore, we propose an approach to incorporate varying degrees of correlation and suggest a practical variant of Rand-Proj-Spatial when the correlation information is not available to the server. Experiments on real-world distributed optimization tasks showcase the superior performance of Rand-Proj-Spatial compared to Rand-k-Spatial and other more sophisticated sparsification techniques.

2.1 Introduction

In modern machine learning applications, data is naturally distributed across a large number of edge devices or clients. The underlying learning task in such settings is modeled by distributed optimization or the recent paradigm of Federated Learning (FL) [56, 61, 72, 111]. A crucial subtask in distributed learning is for the server to compute the mean of the vectors sent by the clients. In FL, for example, clients run training steps on their local data and once-in-a-while send their local models (or local gradients) to the server, which averages them to compute the new global model. However, with the ever-increasing size of machine learning models [19, 95], and the limited battery life of the edge clients, communication cost is often the major constraint for the clients. This motivates the problem of (empirical) *distributed mean estimation* (DME) under communication constraints, as illustrated in Figure 2.1. Each of the n clients holds a vector $\mathbf{x}_i \in \mathbb{R}^d$, on which there are no distributional assumptions. Given a communication budget, each client sends a compressed version $\widehat{\mathbf{x}}_i$ of its vector to the server, which utilizes these to compute an estimate of the mean vector $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$.

Quantization and sparsification are two major techniques for reducing the communication costs of DME. Quantization [26, 46, 98, 109] involves compressing each coordinate of the client vector to a given precision and aims to reduce the number of bits to represent each coordinate, achieving a constant reduction in the commu-

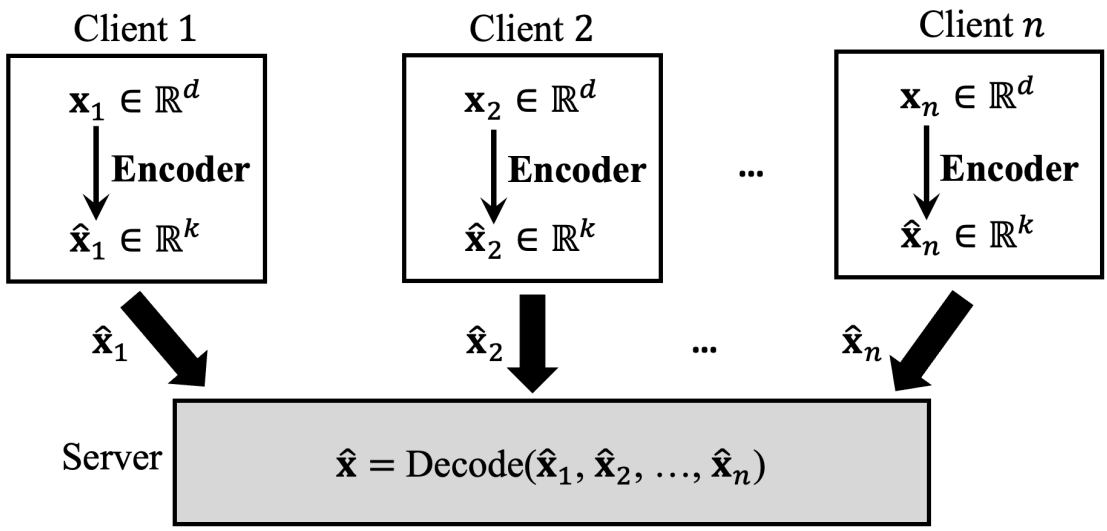

Figure 2.1: The problem of distributed mean estimation under limited communication. Each client $i \in [n]$ encodes its vector $\mathbf{x}_i$ as $\widehat{\mathbf{x}}_i$ and sends this compressed version to the server. The server decodes them to compute an estimate of the true mean $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$.

nication cost. However, the communication cost still remains $\Theta(d)$. Sparsification, on the other hand, aims to reduce the number of coordinates each clinet sends and compresses each client vector to only $k \ll d$ of its coordinates (e.g. Rand-k [60]). As a result, sparsification reduces communication costs more aggressively compared to quantization, achieving better communication efficiency at a cost of only $O(k)$. While in practice, one can use a combination of quantization and sparsification techniques for communication cost reduction, in this work, we focus on the more aggressive sparsification techniques. We call k, the dimension of the vector each client sends to the server, the *per-client* communication budget.

Most existing works on sparsification ignore the potential correlation (or similarity) among the client vectors, which often exists in practice. For example, the data of a specific client in federated learning can be similar to that of multiple clients. Hence, it is reasonable to expect their models (or gradients) to be similar as well. To the best of our knowledge, [52] is the first work to account for *spatial* correlation across individual client vectors. They propose the Rand-k-Spatial family of unbiased estimators, which generalizes Rand-k and achieves a better estimation error in the presence of cross-client correlation. However, their approach is focused only on the

server-side decoding procedure, while the clients do simple Rand-k encoding.

In this work, we consider a more general encoding scheme that directly compresses a vector from $\mathbb{R}^d$ to $\mathbb{R}^k$ using a (random) linear map. The encoded vector consists of k linear combinations of the original coordinates. Intuitively, this has a higher chance of capturing the large-magnitude coordinates ("heavy hitters") of the vector than randomly sampling k out of the d coordinates (Rand-k), which is crucial for the estimator to recover the true mean vector. For example, consider a vector where only a few coordinates are heavy hitters. For small k, Rand-k has a decent chance of missing all the heavy hitters. But with a linear-maps-based general encoding procedure, the large coordinates are more likely to be encoded in the linear measurements, resulting in a more accurate estimator of the mean vector. Guided by this intuition, we ask:

Can we design an improved joint encoding-decoding scheme that utilizes the correlation information and achieves an improved estimation error?

One naïve solution is to apply the same random rotation matrix $\boldsymbol{G} \in \mathbb{R}^{d\times d}$ to each client vector, before applying Rand-k or Rand-k-Spatial encoding. Indeed, such preprocessing is applied to improve the estimator using quantization techniques on heterogeneous vectors [98, 100]. However, as we see in Appendix A.1.1, for sparsification, we can show that this leads to no improvement. But what happens if every client uses a different random matrix, or applies a random $k \times d$-dimensional linear map? How to design the corresponding decoding procedure to leverage cross-client correlation? As there is no way for one to directly apply the decoding procedure of Rand-k-Spatial in such cases. To answer these questions, we propose the Rand-Proj-Spatial family estimator. We propose a flexible encoding procedure in which each client applies its own random linear map to encode the vector. Further, our novel decoding procedure can better leverage cross-client correlation. The resulting mean estimator generalizes and improves over the Rand-k-Spatial family estimator.

Next, we discuss some reasonable restrictions we expect our mean estimator to obey. 1) *Unbiased*. An unbiased mean estimator is theoretically more convenient compared to a biased one [50]. 2) *Non-adaptive*. We focus on an encoding procedure that does not depend on the actual client data, as opposed to the *adaptive* ones, e.g.

Rand-k with vector-based sampling probability [60, 114]. Designing a data-adaptive encoding procedure is computationally expensive as this might require using an iterative procedure to find out the sampling probabilities [60]. In practice, however, clients often have limited computational power compared to the server. Further, as discussed earlier, mean estimation is often a subroutine in more complicated tasks. For applications with streaming data [83], the additional computational overhead of adaptive schemes is challenging to maintain. Note that both Rand-k and Rand-k-Spatial family estimator [52] are *unbiased* and *non-adaptive*.

In this paper, we focus on the severely communication-constrained case $nk \leq d$, when the server receives very limited information about any single client vector. If $nk \gg d$, we see in Appendix A.1.2 that the cross-client information has no additional advantage in terms of improving the mean estimate under both Rand-k-Spatial or Rand-Proj-Spatial, with different choices of random linear maps. Furthermore, when $nk \gg d$, the performance of both the estimators converges to that of Rand-k. Intuitively, this means when the server receives sufficient information regarding the client vectors, it does not need to leverage cross-client correlation to improve the mean estimator.

Our contributions can be summarized as follows:

1. We propose the Rand-Proj-Spatial family estimator with a more flexible encoding-decoding procedure, which can better leverage the cross-client correlation information to achieve a more general and improved mean estimator compared to existing ones.
2. We show the benefit of using Subsampled Randomized Hadamard Transform (SRHT) as the random linear maps in Rand-Proj-Spatial in terms of better mean estimation error (MSE). We theoretically analyze the case when the correlation information is known at the server (see Theorems 2.4.3, 2.4.4 and Section 2.4.3). Further, we propose a practical configuration called Rand-Proj-Spatial(Avg) when the correlation is unknown.
3. We conduct experiments on common distributed optimization tasks, and demonstrate the superior performance of Rand-Proj-Spatial compared to existing sparsification techniques.

2.2 Related Work

Quantization and Sparsification. Commonly used techniques to achieve communication efficiency are quantization, sparsification, or more generic compression schemes, which generalize the former two [14]. Quantization involves either representing each coordinate of the vector by a small number of bits [5, 15, 26, 88, 98, 109], or more involved vector quantization techniques [39, 93]. Sparsification [6, 58, 91, 97, 114], on the other hand, involves communicating a small number $k < d$ of coordinates, to the server. Common protocols include Rand-k [60], sending k uniformly randomly selected coordinates; Top-k [92], sending the k largest magnitude coordinates; and a combination of the two [11]. Some recent works, with a focus on distributed learning, further refine these communication-saving mechanisms [84] by incorporating temporal correlation or error feedback [50, 58].

Distributed Mean Estimation (DME). DME has wide applications in distributed optimization and FL. Most of the existing literature on DME either considers statistical mean estimation [40, 123], assuming that the data across clients is generated i.i.d. according to the same distribution, or empirical mean estimation [20, 52, 60, 71, 98, 108, 110], without making any distributional assumptions on the data. A recent line of work on empirical DME considers applying additional information available to the server, to further improve the mean estimate. This side information includes cross-client correlation [52, 100], or the memory of the past updates sent by the clients [66].

Subsampled Randomized Hadamard Transformation (SRHT). SRHT was introduced for random dimensionality reduction using sketching [4, 64, 105]. Common applications of SRHT include faster computation of matrix problems, such as low-rank approximation [9, 18], and machine learning tasks, such as ridge regression [70], and least square problems [48, 63, 102]. SRHT has also been applied to improve communication efficiency in distributed optimization [51] and FL [47, 89].

2.3 Preliminaries

Notation. We use bold lowercase (uppercase) letters, e.g. $\mathbf{x}$ ($\boldsymbol{G}$) to denote vectors (matrices). $\mathbf{e}_j \in \mathbb{R}^d$, for $j \in [d]$, denotes the j-th canonical basis vector. $\|\cdot\|_2$ denotes

the Euclidean norm. For a vector $\mathbf{x}$, $\mathbf{x}(j)$ denotes its j-th coordinate. Given integer m, we denote by $[m]$ the set $\{1, 2, \ldots, m\}$.

Problem Setup. Consider n geographically separated clients coordinated by a central server. Each client $i \in [n]$ holds a vector $\mathbf{x}_i \in \mathbb{R}^d$, while the server wants to estimate the mean vector $\bar{\mathbf{x}} \triangleq \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Given a per-client communication budget of $k \in [d]$, each client i computes $\widehat{\mathbf{x}}_i$ and sends it to the central server. $\widehat{\mathbf{x}}_i$ is an approximation of $\mathbf{x}_i$ that belongs to a random k-dimensional subspace. Each client also sends a random seed to the server, which conveys the subspace information, and can usually be communicated using a negligible amount of bits. Having received the encoded vectors $\{\widehat{\mathbf{x}}_i\}_{i=1}^n$, the server then computes $\widehat{\mathbf{x}} \in \mathbb{R}^d$, an estimator of $\bar{\mathbf{x}}$. We consider the severely communication-constrained setting where $nk \leq d$, when only a limited amount of information about the client vectors is seen by the server.

Error Metric. We measure the quality of the decoded vector $\widehat{\mathbf{x}}$ using the Mean Squared Error (MSE) $\mathbb{E}\left[\|\widehat{\mathbf{x}} - \bar{\mathbf{x}}\|_2^2\right]$, where the expectation is with respect to all the randomness in the encoding-decoding scheme. Our goal is to design an encoding-decoding algorithm to achieve an unbiased estimate $\widehat{\mathbf{x}}$ (i.e. $\mathbb{E}[\widehat{\mathbf{x}}] = \bar{\mathbf{x}}$) that minimizes the MSE, given the per-client communication budget k. To consider an example, in rand-k sparsification, each client sends randomly selected k out of its d coordinates to the server. The server then computes the mean estimate as $\widehat{\mathbf{x}}^{(\text{Rand-}k)} = \frac{1}{n}\frac{d}{k}\sum_{i=1}^n \widehat{\mathbf{x}}_i$. By [52, Lemma 1], the MSE of Rand-k sparsification is given by

$$\mathbb{E}\left[\|\widehat{\mathbf{x}}^{(\text{Rand-}k)} - \bar{\mathbf{x}}\|_2^2\right] = \frac{1}{n^2}\left(\frac{d}{k} - 1\right)\sum_{i=1}^n \|\mathbf{x}_i\|_2^2 \tag{2.1}$$

The Rand-k-Spatial Family Estimator. For large values of $\frac{d}{k}$, the Rand-k MSE in Eq. 2.1 can be prohibitive. [52] proposed the Rand-k-Spatial family estimator, which achieves an improved MSE, by leveraging the knowledge of the correlation between client vectors at the server. The encoded vectors $\{\widehat{\mathbf{x}}_i\}$ are the same as in Rand-k. However, the j-th coordinate of the decoded vector is given as

$$\widehat{\mathbf{x}}^{(\text{Rand-}k\text{-Spatial})}(j) = \frac{1}{n}\frac{\bar{\beta}}{T(M_j)}\sum_{i=1}^n \widehat{\mathbf{x}}_i(j) \tag{2.2}$$

Here, $T : \mathbb{R} \to \mathbb{R}$ is a pre-defined transformation function of M_j, the number of clients

which sent their j-th coordinate, and $\bar{\beta}$ is a normalization constant to ensure $\widehat{\mathbf{x}}$ is an unbiased estimator of $\mathbf{x}$. The resulting MSE is given by

$$\mathbb{E}\left[\|\widehat{\mathbf{x}}^{(\text{Rand-}k\text{-Spatial})} - \bar{\mathbf{x}}\|_2^2\right] = \frac{1}{n^2}\left(\frac{d}{k} - 1\right)\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2 + \left(c_1\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2 - c_2\sum_{i=1}^{n}\sum_{l\neq i}\langle\mathbf{x}_i, \mathbf{x}_l\rangle\right) \tag{2.3}$$

where c_1, c_2 are constants dependent on n, d, k and T, but independent of client vectors $\{\mathbf{x}_i\}_{i=1}^n$. When the client vectors are orthogonal, i.e., $\langle\mathbf{x}_i, \mathbf{x}_l\rangle = 0$, for all $i \neq l$, [52] show that with appropriately chosen T, the MSE in Eq. 2.3 reduces to Eq. 2.1. However, if there exists a positive correlation between the vectors, the MSE in Eq. 2.3 is strictly smaller than that for Rand-k Eq. 2.1.

2.4 The Rand-Proj-Spatial Family Estimator

While the Rand-k-Spatial family estimator proposed in [52] focuses only on improving the decoding at the server, we consider a more general encoding-decoding scheme. Rather than simply communicating k out of the d coordinates of its vector $\mathbf{x}_i$ to the server, client i applies a (random) linear map $\boldsymbol{G}_i \in \mathbb{R}^{k\times d}$ to $\mathbf{x}_i$ and sends $\widehat{\mathbf{x}}_i = \boldsymbol{G}_i\mathbf{x}_i \in \mathbb{R}^k$ to the server. The decoding process on the server first projects the *encoded* vectors $\{\boldsymbol{G}_i\mathbf{x}_i\}_{i=1}^n$ back to the d-dimensional space and then forms an estimate $\widehat{\mathbf{x}}$. We motivate our new decoding procedure with the following regression problem:

$$\widehat{\mathbf{x}}^{(\text{Rand-Proj})} = \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{n}\|\boldsymbol{G}_i\mathbf{x} - \boldsymbol{G}_i\mathbf{x}_i\|_2^2 \tag{2.4}$$

To understand the motivation behind Eq. 2.4, first consider the special case where $\boldsymbol{G}_i = \boldsymbol{I}_d$ for all $i \in [n]$, that is, the clients communicate their vectors without compressing. The server can then exactly compute the mean $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Equivalently, $\bar{\mathbf{x}}$ is the solution of $\operatorname{argmin}_{\mathbf{x}}\sum_{i=1}^n\|\mathbf{x} - \mathbf{x}_i\|_2^2$. In the more general setting, we require that the mean estimate $\widehat{\mathbf{x}}$ when encoded using the map $\boldsymbol{G}_i$, should be "close" to the encoded vector $\boldsymbol{G}_i\mathbf{x}_i$ originally sent by client i, for all clients $i \in [n]$.

We note the above intuition can also be translated into different regression

problems to motivate the design of the new decoding procedure. We discuss in Appendix A.2.2 intuitive alternatives which, unfortunately, either do not enable the usage of cross-client correlation information, or do not use such information effectively. We choose the formulation in Eq. 2.4 due to its analytical tractability and its direct relevance to our target error metric MSE. We note that it is possible to consider the problem in Eq. 2.4 in the other norms, such as the sum of ℓ_2 norms (without the squares) or the ℓ_∞ norm. We leave this as a future direction to explore.

The solution to Eq. 2.4 is given by $\widehat{\mathbf{x}}^{(\text{Rand-Proj})} = (\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i)^\dagger \sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i \mathbf{x}_i$, where $\dagger$ denotes the Moore-Penrose pseudo inverse [44]. However, while $\widehat{\mathbf{x}}^{(\text{Rand-Proj})}$ minimizes the error of the regression problem, our goal is to design an *unbiased* estimator that also improves the MSE. Therefore, we make the following two modifications to $\widehat{\mathbf{x}}^{(\text{Rand-Proj})}$: First, to ensure that the mean estimate is unbiased, we scale the solution by a normalization factor $\bar{\beta}$[2]. Second, to incorporate varying degrees of correlation among the clients, we propose to apply a scalar transformation function $T : \mathbb{R} \to \mathbb{R}$ to each of the eigenvalues of $\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i$. The resulting Rand-Proj-Spatial family estimator is given by

$$\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial})} = \bar{\beta}\Big(T\big(\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i\big)\Big)^\dagger \sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i \mathbf{x}_i \tag{2.5}$$

Though applying the transformation function T in Rand-Proj-Spatial requires computing the eigendecomposition of $\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i$. However, this happens only at the server, which has more computational power than the clients. Next, we observe that for appropriate choice of $\{\boldsymbol{G}_i\}_{i=1}^n$, the Rand-Proj-Spatial family estimator reduces to the Rand-k-Spatial family estimator [52].

Lemma 2.4.1 (Recovering Rand-k-Spatial)**.** *Suppose client i generates a subsampling matrix $\boldsymbol{E}_i = \begin{bmatrix}\mathbf{e}_{i_1}, & \ldots, & \mathbf{e}_{i_k}\end{bmatrix}^\top$, where $\{\mathbf{e}_j\}_{j=1}^d$ are the canonical basis vectors, and $\{i_1, \ldots, i_k\}$ are sampled from $\{1, \ldots, d\}$ without replacement. The encoded vectors are given as $\widehat{\mathbf{x}}_i = \boldsymbol{E}_i \mathbf{x}_i$. Given a function T, $\widehat{\mathbf{x}}$ computed as in Eq. 2.5 recovers the Rand-k-Spatial estimator.*

The proof details are in Appendix A.3.5. We discuss the choice of T and how it compares to Rand-k-Spatial in detail in Section 2.4.3.

[2]We show that it suffices for $\bar{\beta}$ to be a scalar in Appendix A.2.1.

Remark 2.4.2. *In the simple case when $\boldsymbol{G}_i$'s are subsampling matrices (as in Rand-k-Spatial [52]), the j-th diagonal entry of $\sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$, M_j conveys the number of clients which sent the j-th coordinate. Rand-k-Spatial incorporates correlation among client vectors by applying a function T to M_j. Intuitively, it means scaling different coordinates differently. This is in contrast to Rand-k, which scales all the coordinates by d/k. In our more general case, we apply a function T to the eigenvalues of $\sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$ to similarly incorporate correlation in Rand-Proj-Spatial.*

To showcase the utility of the Rand-Proj-Spatial family estimator, we propose to set the random linear maps $\boldsymbol{G}_i$ to be scaled Subsampled Randomized Hadamard Transform (SRHT, e.g. [105]). Assuming d to be a power of 2, the linear map $\boldsymbol{G}_i$ is given as

$$\boldsymbol{G}_i = \frac{1}{\sqrt{d}} \boldsymbol{E}_i \boldsymbol{H} \boldsymbol{D}_i \in \mathbb{R}^{k \times d} \tag{2.6}$$

where $\boldsymbol{E}_i \in \mathbb{R}^{k \times d}$ is the subsampling matrix, $\boldsymbol{H} \in \mathbb{R}^{d \times d}$ is the (deterministic) Hadamard matrix and $\boldsymbol{D}_i \in \mathbb{R}^{d \times d}$ is a diagonal matrix with independent Rademacher random variables as its diagonal entries. We choose SRHT due to its superior performance compared to other random matrices. Other possible choices of random matrices for Rand-Proj-Spatial estimator include sketching matrices commonly used for dimensionality reduction, such as Gaussian [104, 116], row-normalized Gaussian, and Count Sketch [75], as well as error-correction coding matrices, such as Low-Density Parity Check (LDPC) [38] and Fountain Codes [94]. However, in the absence of correlation between client vectors, all these matrices suffer a higher MSE.

In the following, we first compare the MSE of Rand-Proj-Spatial with SRHT against Rand-k and Rand-k-Spatial in two extreme cases: when all the client vectors are identical, and when all the client vectors are orthogonal to each other. In both cases, we highlight the transformation function T used in Rand-Proj-Spatial (Eq. 2.5) to incorporate the knowledge of cross-client correlation. We define

$$\mathcal{R} := \frac{\sum_{i=1}^{n} \sum_{l \neq i} \langle \mathbf{x}_i, \mathbf{x}_l \rangle}{\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2} \tag{2.7}$$

to measure the correlation between the client vectors. Note that $\mathcal{R} \in [-1, n-1]$. $\mathcal{R} = 0$ implies all client vectors are orthogonal, while $\mathcal{R} = n - 1$ implies identical

client vectors.

2.4.1 Case I: Identical Client Vectors ($\mathcal{R} = n - 1$)

When all the client vectors are identical ($\mathbf{x}_i \equiv \mathbf{x}$), [52] showed that setting the transformation T to identity, i.e., $T(m) = m$, for all m, leads to the minimum MSE in the Rand-k-Spatial family of estimators. The resulting estimator is called Rand-k-Spatial (Max). Under the same setting, using the same transformation T in Rand-Proj-Spatial with SRHT, the decoded vector in Eq. 2.5 simplifies to

$$\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial})} = \bar{\beta}\Big(\sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i\Big)^{\dagger} \sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i \mathbf{x} = \bar{\beta} \boldsymbol{S}^{\dagger} \boldsymbol{S} \mathbf{x}, \tag{2.8}$$

where $\boldsymbol{S} := \sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$. By construction, $\text{rank}(\boldsymbol{S}) \leq nk$, and we focus on the case $nk \leq d$.

Limitation of Subsampling matrices. As mentioned above, with $\boldsymbol{G}_i = \boldsymbol{E}_i, \forall\ i \in [n]$, we recover the Rand-$k$-Spatial family of estimators. In this case, $\boldsymbol{S}$ is a diagonal matrix, where each diagonal entry $\boldsymbol{S}_{jj} = M_j$, $j \in [d]$. M_j is the number of clients which sent their j-th coordinate to the server. To ensure $\text{rank}(\boldsymbol{S}) = nk$, we need $\boldsymbol{S}_{jj} \leq 1, \forall j$, i.e., each of the d coordinates is sent by *at most* one client. If all the clients sample their matrices $\{\boldsymbol{E}_i\}_{i=1}^n$ independently, this happens with probability $\frac{\binom{d}{nk}}{\binom{d}{k}^n}$. As an example, for $k = 1$, $\text{Prob}(\text{rank}(\boldsymbol{S}) = n) = \frac{\binom{d}{n}}{d^n} \leq \frac{1}{n!}$ (because $\frac{d^n}{n^n} \leq \binom{d}{n} \leq \frac{d^n}{n!}$). Therefore, to guarantee that $\boldsymbol{S}$ is full-rank, each client would need the subsampling information of all the other clients. This not only requires additional communication but also has serious privacy implications. Essentially, the limitation with subsampling matrices $\boldsymbol{E}_i$ is that the eigenvectors of $\boldsymbol{S}$ are restricted to be canonical basis vectors $\{\mathbf{e}_j\}_{j=1}^d$. Generalizing $\boldsymbol{G}_i$'s to general rank k matrices relaxes this constraint and hence we can ensure that $\boldsymbol{S}$ is full-rank with high probability. In the next result, we show the benefit of choosing $\boldsymbol{G}_i$ as SRHT matrices. We call the resulting estimator Rand-Proj-Spatial(Max).

Theorem 2.4.3 (MSE under Full Correlation)**.** *Consider n clients, each holding the same vector $\mathbf{x} \in \mathbb{R}^d$. Suppose we set $T(\lambda) = \lambda$, $\bar{\beta} = \frac{d}{k}$ in Eq. 2.5, and the random linear map $\boldsymbol{G}_i$ at each client to be an SRHT matrix. Let δ be the probability that*

$\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$ does not have full rank. Then, for $nk \leq d$,

$$\mathbb{E}\left[\|\widehat{\mathbf{x}}^{(\textit{Rand-Proj-Spatial(Max)})} - \bar{\mathbf{x}}\|_2^2\right] \leq \left[\frac{d}{(1-\delta)nk + \delta k} - 1\right]\|\mathbf{x}\|_2^2 \tag{2.9}$$

The proof details are in Appendix A.3.1. To compare the performance of Rand-Proj-Spatial(Max) against Rand-k, we show in Appendix A.3.2 that for $n \geq 2$, as long as $\delta \leq \frac{2}{3}$, the MSE of Rand-Proj-Spatial(Max) is less than that of Rand-k. Furthermore, in Appendix A.3.3 we empirically demonstrate that with $d \in \{32, 64, 128, \ldots, 1024\}$ and different values of $nk \leq d$, the rank of $\boldsymbol{S}$ is full with high probability, i.e., $\delta \approx 0$. This implies $\mathbb{E}[\|\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial(Max)})} - \bar{\mathbf{x}}\|_2^2] \approx (\frac{d}{nk} - 1)\|\mathbf{x}\|_2^2$.

Furthermore, since setting $\boldsymbol{G}_i$ as SRHT significantly increases the probability of recovering nk coordinates of $\mathbf{x}$, the MSE of Rand-Proj-Spatial with SRHT (Eq. 2.4.3) is strictly less than that of Rand-k-Spatial (Eq. 2.3). We also compare the MSEs of the three estimators in Figure 2.2 in the following setting: $\|\mathbf{x}\|_2 = 1$, $d = 1024, n \in \{10, 20, 50, 100\}$ and small k values such that $nk < d$.

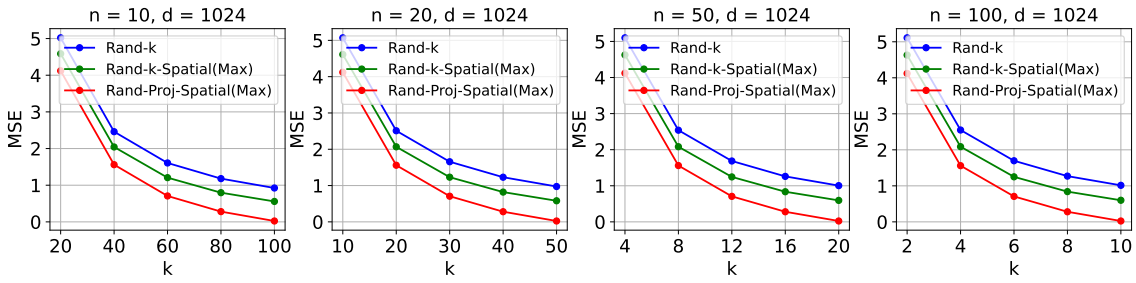

Figure 2.2: MSE comparison of Rand-k, Rand-k-Spatial(Max) and Rand-Proj-Spatial(Max) estimators, when all clients have identical vectors (maximum inter-client correlation).

2.4.2 Case II: Orthogonal Client Vectors ($\mathcal{R} = 0$)

When all the client vectors are orthogonal to each other, [52] showed that Rand-k has the lowest MSE among the Rand-k-Spatial family of decoders. We show in the next result that if we set the random linear maps $\boldsymbol{G}_i$ at client i to be SRHT, and choose the fixed transformation $T \equiv 1$ as in [52], Rand-Proj-Spatial achieves the same MSE as that of Rand-k.

Theorem 2.4.4 (MSE under No Correlation)**.** *Consider n clients, each holding a vector $\mathbf{x}_i \in \mathbb{R}^d$, $\forall i \in [n]$. Suppose we set $T \equiv 1$, $\bar{\beta} = \frac{d^2}{k}$ in Eq. 2.5, and the random linear map $\boldsymbol{G}_i$ at each client to be an SRHT matrix. Then, for $nk \leq d$,*

$$\mathbb{E}\left[\|\widehat{\mathbf{x}}^{(\textit{Rand-Proj-Spatial})} - \bar{\mathbf{x}}\|_2^2\right] = \frac{1}{n^2}\left(\frac{d}{k} - 1\right)\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2. \tag{2.10}$$

The proof details are in Appendix A.3.4. Theorem 2.4.4 above shows that with zero correlation among client vectors, Rand-Proj-Spatial achieves the same MSE as that of Rand-k.

2.4.3 Incorporating Varying Degrees of Correlation

In practice, it unlikely that all the client vectors are either identical or orthogonal to each other. In general, there is some "imperfect" correlation among the client vectors, i.e., $\mathcal{R} \in (0, n-1)$. Given correlation level $\mathcal{R}$, [52] shows that the estimator from the Rand-k-Spatial family that minimizes the MSE is given by the following transformation.

$$T(m) = 1 + \frac{\mathcal{R}}{n-1}(m-1) \tag{2.11}$$

Recall from Section 2.4.1 (Section 2.4.2) that setting $T(m) = 1$ ($T(m) = m$) leads to the estimator among the Rand-k-Spatial family that minimizes MSE when there is zero (maximum) correlation among the client vectors. We observe the function T defined in Eq. 2.11 essentially interpolates between the two extreme cases, using the normalized degree of correlation $\frac{\mathcal{R}}{n-1} \in [-\frac{1}{n-1}, 1]$ as the weight. This motivates us to apply the same function T defined in Eq. 2.11 on the eigenvalues of $\boldsymbol{S} = \sum_{i=1}^{n}\boldsymbol{G}_i^T\boldsymbol{G}_i$ in Rand-Proj-Spatial. As we shall see in our results, the resulting Rand-Proj-Spatial family estimator improves over the MSE of both Rand-k and Rand-k-Spatial family estimator.

We note that deriving a closed-form expression of MSE for Rand-Proj-Spatial with SRHT in the general case with the transformation function T (Eq. 2.11) is hard (we elaborate on this in Appendix A.2.3), as this requires a closed form expression for the non-asymptotic distributions of eigenvalues and eigenvectors of the random

matrix $\boldsymbol{S}$. To the best of our knowledge, previous analyses of SRHT, for example in [4, 63, 64, 65, 105], rely on the asymptotic properties of SRHT, such as the limiting eigen spectrum, or concentration bounds on the singular values, to derive asymptotic or approximate guarantees. However, to analyze the MSE of Rand-Proj-Spatial, we need an exact, non-asymptotic analysis of the eigenvalues and eigenvectors distribution of SRHT. Given the apparent intractability of the theoretical analysis, we compare the MSE of Rand-Proj-Spatial, Rand-k-Spatial, and Rand-k via simulations.

Simulations. In each experiment, we first simulate $\bar{\beta}$ in Eq. 2.5, which ensures our estimator is unbiased, based on 1000 random runs. Given the degree of correlation $\mathcal{R}$, we then compute the squared error, i.e. $\|\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial})} - \bar{\mathbf{x}}\|_2^2$, where Rand-Proj-Spatial has $\boldsymbol{G}_i$ as SRHT matrix (Eq. 2.6) and T as in Eq. 2.11. We plot the average over 1000 random runs as an approximation to MSE. Each client holds a d-dimensional base vector $\mathbf{e}_j$ for some $j \in [d]$, and so two clients either hold the same or orthogonal vectors. We control the degree of correlation $\mathcal{R}$ by changing the number of clients which hold the same vector. We consider $d = 1024$, $n \in \{21, 51\}$. We consider positive correlation values, where $\mathcal{R}$ is chosen to be linearly spaced within $[0, n-1]$. Hence, for $n = 21$, we use $\mathcal{R} \in \{4, 8, 12, 16\}$ and for $n = 51$, we use $\mathcal{R} \in \{10, 20, 30, 40\}$. All results are presented in Figure 2.3. As expected, given $\mathcal{R}$, Rand-Proj-Spatial consistently achieves a lower MSE than the lowest possible MSE from the Rand-k-Spatial family decoder. Additional results with different values of n, d, k, including the setting $nk \ll d$, can be found in Appendix A.2.4.

A Practical Configuration. In reality, it is hard to know the correlation information $\mathcal{R}$ among the client vectors. [52] uses the transformation function which interpolates to the middle point between the full correlation and no correlation cases, such that $T(m) = 1 + \frac{n}{2}\frac{m-1}{n-1}$. Rand-$k$-Spatial with such T is called Rand-k-Spatial(Avg). Following this approach, we evaluate Rand-Proj-Spatial with SRHT using this T, and call it Rand-Proj-Spatial(Avg) in practical settings (see Figure 2.4).

2.5 Experiments

We consider three practical distributed optimization tasks for evaluation: distributed power iteration, distributed k-means and distributed linear regression. We compare

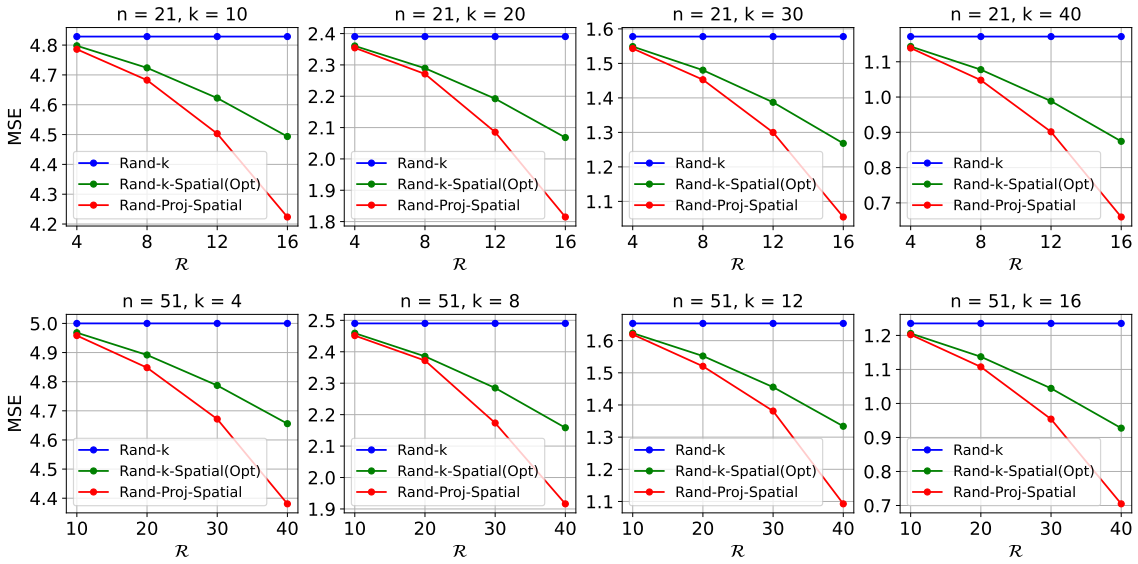

Figure 2.3: MSE comparison of estimators Rand-k, Rand-k-Spatial(Opt), Rand-Proj-Spatial, given the degree of correlation $\mathcal{R}$. Rand-k-Spatial(Opt) denotes the estimator that gives the lowest possible MSE from the Rand-k-Spatial family. We consider $d = 1024$, number of clients $n \in \{21, 51\}$, and k values such that $nk < d$. In each plot, we fix n, k, d and vary the degree of positive correlation $\mathcal{R}$. The y-axis represents MSE. Notice since each client has a fixed $\|\mathbf{x}_i\|_2 = 1$, and Rand-k does not leverage cross-client correlation, the MSE of Rand-k in each plot remains the same for different $\mathcal{R}$.

Rand-Proj-Spatial(Avg) against Rand-k, Rand-k-Spatial(Avg), and two more sophisticated but widely used sparsification schemes: non-uniform coordinate-wise gradient sparsification [114] (we call it Rand-k(Wangni)) and the Induced compressor with Rand-k + Top-k [50]. The results are presented in Figure 2.4.

Dataset. For both distributed power iteration and distributed k-means, we use the test set of the `Fashion-MNIST` dataset [120] consisting of 10000 samples. The original images from `Fashion-MNIST` are 28×28 in size. We preprocess and resize each image to be 32×32. Resizing images to have their dimension as a power of 2 is a common technique used in computer vision to accelerate the convolution operation. We use the `UJIndoor` dataset [3] for distributed linear regression. We subsample 10000 data points, and use the first 512 out of the total 520 features on signals of phone calls. The task is to predict the longitude of the location of a phone call. In all the

[3] https://archive.ics.uci.edu/ml/datasets/ujiindoorloc

experiments in Figure 2.4, the datasets are split IID across the clients via random shuffling. In Appendix A.4.1, we have additional results for non-IID data split across the clients.

Setup and Metric. Recall that n denotes the number of clients, k the per-client communication budget, and d the vector dimension. For Rand-Proj-Spatial, we use the first 50 iterations to estimate $\bar{\beta}$ (see Eq. 2.5). Note that $\bar{\beta}$ only depends on n, k, d, and T (the transformation function in Eq. 2.5), but is independent of the dataset. We repeat the experiments across 10 independent runs, and report the mean MSE (solid lines) and one standard deviation (shaded regions) for each estimator. For each task, we plot the squared error of the mean estimator $\widehat{\mathbf{x}}$, i.e., $\|\widehat{\mathbf{x}} - \bar{\mathbf{x}}\|_2^2$, and the values of the task-specific loss function, detailed below.

Tasks and Settings:

1. Distributed power iteration. We estimate the principle eigenvector of the covariance matrix, with the dataset (`Fashion-MNIST`) distributed across the n clients. In each iteration, each client computes a local principle eigenvector estimate based on a single power iteration and sends an encoded version to the server. The server then computes a global estimate and sends it back to the clients. The task-specific loss here is $\|\mathbf{v}_t - \mathbf{v}_{top}\|_2$, where $\mathbf{v}_t$ is the global estimate of the principal eigenvector at iteration t, and $\mathbf{v}_{top}$ is the true principle eigenvector.

2. Distributed k-means. We perform k-means clustering [10] with the data distributed across n clients (`Fashion-MNIST`, 10 classes) using Lloyd's algorithm. At each iteration, each client performs a single iteration of k-means to find its local centroids and sends the encoded version to the server. The server then computes an estimate of the global centroids and sends them back to the clients. We report the average squared mean estimation error across 10 clusters, and the k-means loss, i.e., the sum of the squared distances of the data points to the centroids.

For both distributed power iterations and distributed k-means, we run the experiments for 30 iterations and consider two different settings: $n = 10, k = 102$ and $n = 50, k = 20$.

3. Distributed linear regression. We perform linear regression on the `UJIndoor` dataset distributed across n clients using SGD. At each iteration, each client computes a local gradient and sends an encoded version to the server. The server computes a global estimate of the gradient, performs an SGD step, and sends

the updated parameter to the clients. We run the experiments for 50 iterations with learning rate 0.001. The task-specific loss is the linear regression loss, i.e. empirical mean squared error. To have a proper scale that better showcases the difference in performance of different estimators, we plot the results starting from the 10th iteration.

Results. It is evident from Figure 2.4 that Rand-Proj-Spatial(Avg), our estimator with the practical configuration T (see Section 2.4.3) that does not require the knowledge of the actual degree of correlation among clients, consistently outperforms the other estimators in all three tasks. Additional experiments for the three tasks are included in Appendix A.4.1. Furthermore, we present the wall-clock time to encode and decode client vectors using different sparsification schemes in Figure 2.5. Though Rand-Proj-Spatial(Avg) has the longest decoding time, the encoding time of Rand-Proj-Spatial(Avg) is less than that of the *adaptive* Rand-k(Wangni) sparsifier. In practice, the server has more computational power than the clients and hence can afford a longer decoding time. Therefore, it is more important to have efficient encoding procedures.

2.6 Limitations

We note two practical limitations of the proposed Rand-Proj-Spatial.

1) Computation Time of Rand-Proj-Spatial. The encoding time of Rand-Proj-Spatial is $O(kd)$, while the decoding time is $O(d^2 \cdot nk)$. The computation bottleneck in decoding is computing the eigendecomposition of the $d \times d$ matrix $\sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$ of rank at most nk. Improving the computation time for both the encoding and decoding schemes is an important direction for future work.

2) Perfect Shared Randomness. It is common to assume perfect shared randomness between the server and the clients in distributed settings [124]. However, to perfectly simulate randomness using Pseudo Random Number Generator (PRNG), at least $\log_2 d$ bits of the seed need to be exchanged in practice. We acknowledge this gap between theory and practice.

2.7 Conclusion

In this work, we propose the Rand-Proj-Spatial estimator, a novel encoding-decoding scheme, for communication-efficient distributed mean estimation. The proposed client-side encoding generalizes and improves the commonly used Rand-k sparsification, by utilizing projections onto general k-dimensional subspaces. On the server side, cross-client correlation is leveraged to improve the approximation error. Compared to existing methods, the proposed scheme consistently achieves better mean estimation error across a variety of tasks. Potential future directions include improving the computation time of Rand-Proj-Spatial and exploring whether the proposed Rand-Proj-Spatial achieves the optimal estimation error among the class of *non-adaptive* estimators, given correlation information. Furthermore, combining sparsification and quantization techniques and deriving such algorithms with the optimal communication cost-estimation error trade-offs would be interesting.

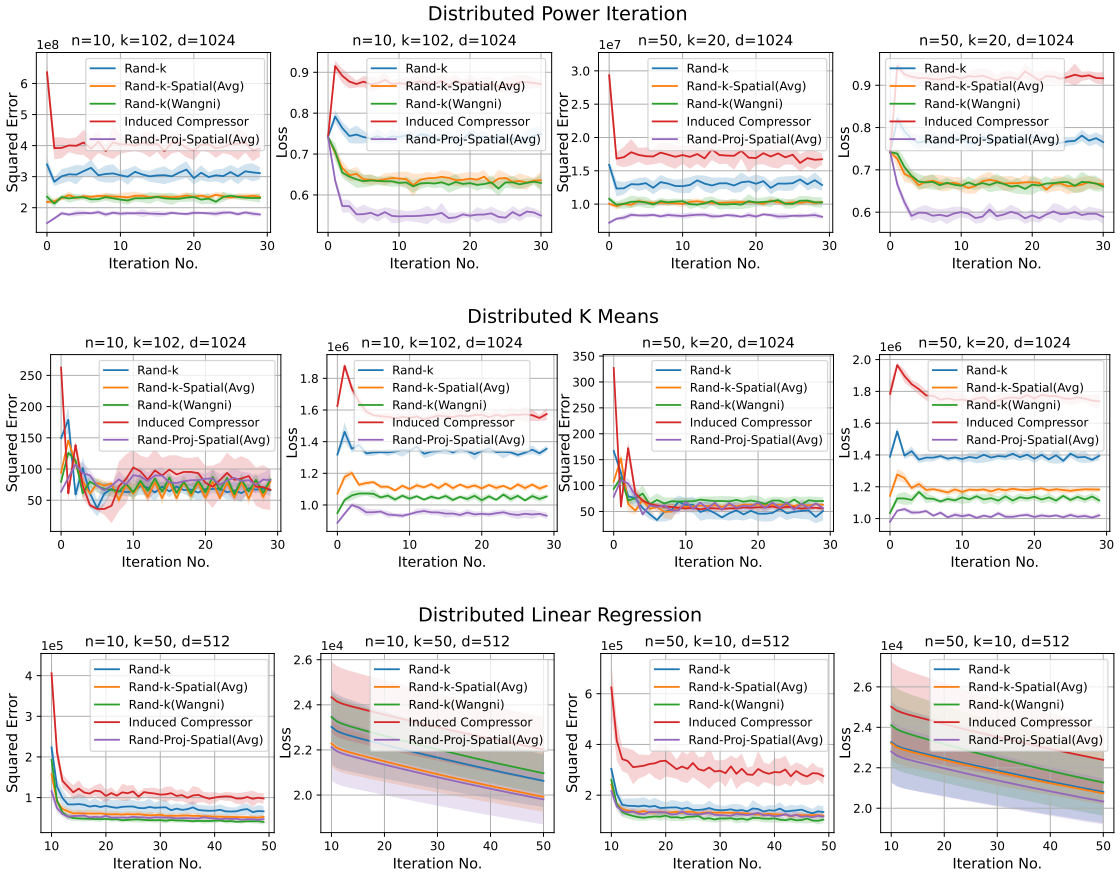

Figure 2.4: Experiment results on three distributed optimization tasks: distributed power iteration, distributed k-means, and distributed linear regression. The first two use the `Fashion-MNIST` dataset with the images resized to 32×32, hence $d = 1024$. Distributed linear regression uses `UJIndoor` dataset with $d = 512$. All the experiments are repeated for 10 random runs, and we report the mean as the solid lines, and one standard deviation using the shaded region. The violet line in the plots represents our proposed Rand-Proj-Spatial(Avg) estimator.

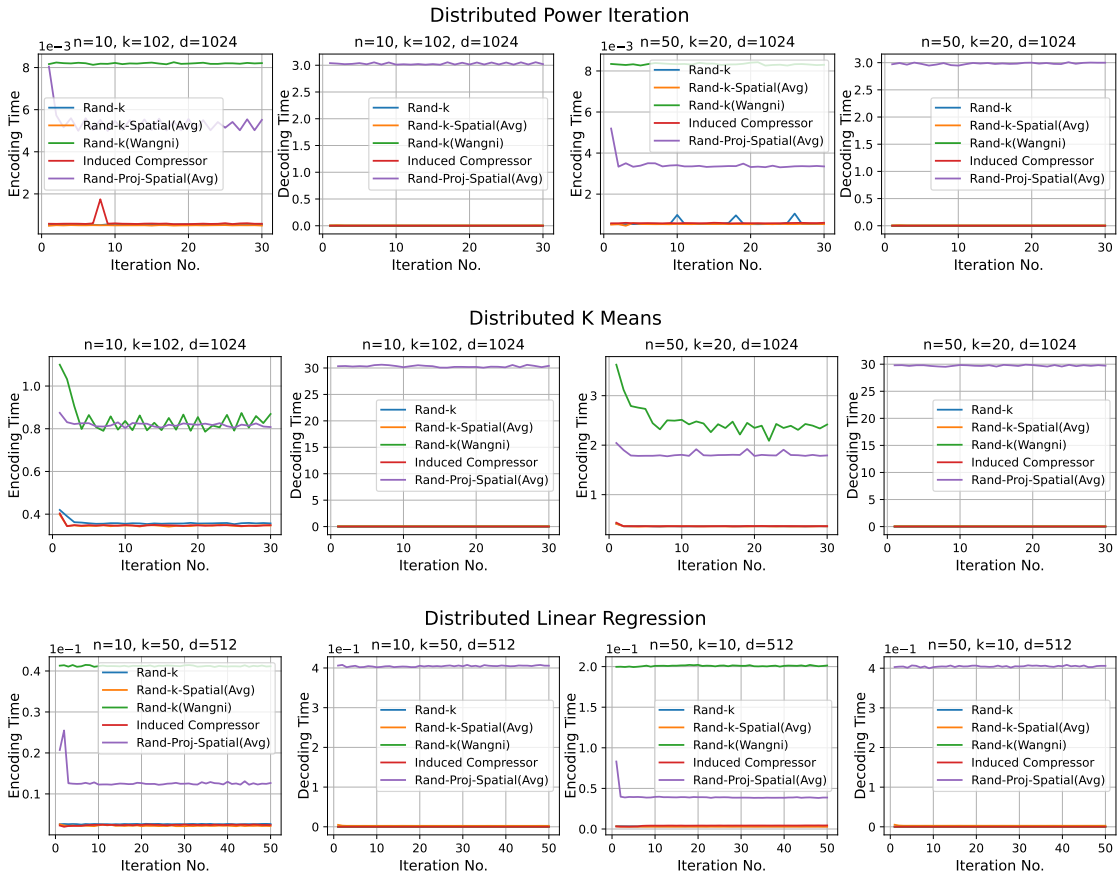

Figure 2.5: The corresponding wall-clock time to encode and decode client vectors (in seconds) using different sparsification schemes, across the three tasks.

Chapter 3

Differential Privacy: Private Majority Ensembling

This chapter is based on the following work:
Shuli Jiang, Richard Zhang, Gauri Joshi. "Optimized Tradeoffs for Private Prediction with Majority Ensembling". Transactions on Machine Learning Research (TMLR), 2024.

Beyond communication efficiency, another critical challenge in distributed learning is ensuring clients' data privacy. A widely used approach to address this challenge is differential privacy—a rigorous mathematical framework that quantifies the risks of leaking sensitive information from clients' data through the outputs, such as the collaboratively trained global model.

In this chapter, we turn our attention to differential privacy, specifically exploring a classical problem in private prediction: the problem of computing an $(m\epsilon, \delta)$-differentially private majority of K (ϵ, Δ)-differentially private algorithms for $1 \leq m \leq K$ and $1 > \delta \geq \Delta \geq 0$. Standard methods such as subsampling or randomized response are widely used, but do they provide optimal privacy-utility tradeoffs? To answer this, we introduce the Data-dependent Randomized Response Majority (DaRRM) algorithm. It is parameterized by a data-dependent noise function γ, and enables efficient utility optimization over the class of all private algorithms, encompassing those standard methods. We show that maximizing the utility of an

$(m\epsilon, \delta)$-private majority algorithm can be computed tractably through an optimization problem for any $m \leq K$ by a novel structural result that reduces the infinitely many privacy constraints into a polynomial set. In some settings, we show that DaRRM provably enjoys a privacy gain of a factor of 2 over common baselines, with fixed utility. Lastly, we demonstrate the strong empirical effectiveness of our first-of-its-kind privacy-constrained utility optimization for ensembling labels for private prediction from private teachers in image classification. Notably, our DaRRM framework with an optimized γ exhibits substantial utility gains when compared against several baselines.

3.1 Introduction

Differential privacy (DP) is a widely applied framework for formally reasoning about privacy leakage when releasing statistics on a sensitive database [24, 33]. Differential privacy protects data privacy by obfuscating algorithmic output, ensuring that query responses look similar on adjacent datasets while preserving utility as much as possible [31].

Privacy in practice often requires aggregating or composing multiple private procedures that are distributed for data or training efficiency. For example, it is common to aggregate multiple private algorithmic or model outputs in methods such as boosting or calibration [90]. In federated learning, model training is distributed across multiple edge devices. Those devices need to send local information, such as labels or gradients [62], to an aggregating server, which is often honest but curious about the local training data. Hence, the output from each model at an edge device needs to be privatized locally before being sent to the server. When translating from a local privacy guarantee to a centralized one, one needs to reason about the composition of the local privacy leakage [81]. Therefore, we formally ask the following:

Problem 3.1.1 (Private Majority Ensembling (Illustrated in Figure 3.1))**.** *Consider $K \geq 1$ (ϵ, Δ)-differentially private mechanisms $M_1, \ldots, M_K$ for K odd. Given a dataset $\mathcal{D}$, each mechanism outputs a binary answer — that is, $M_i : \mathcal{D} \to \{0, 1\}$, $\forall i \in [K]$. Given a privacy* ***allowance*** *$1 \leq m \leq K$, $m \in \mathbb{R}$ and a failure probability $\delta \geq \Delta \geq 0$, $\delta, \Delta \in [0, 1)$, how can one maximize the utility of an $(m\epsilon, \delta)$-differentially private mechanism $\mathcal{A}$ to compute the majority function $g(S_1, S_2, \ldots, S_K)$, where*

$S_i \sim M_i(\mathcal{D})$?

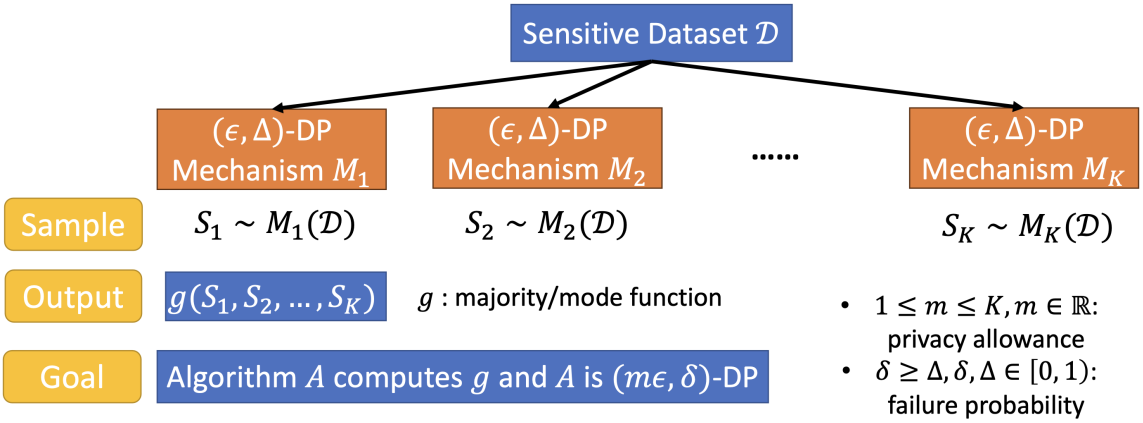

Figure 3.1: An illustration of the problem setting. The inputs are the dataset $\mathcal{D}$ and K (ϵ, Δ)-differentially private mechanisms $M_1, \ldots, M_K$. One draws samples $S_i \sim M_i(\mathcal{D})$ and computes an aggregated output $g(S_1, \ldots, S_K)$ based on all observed samples. Our goal is to design a randomized algorithm $\mathcal{A}$ that approximately computes g and is $(m\epsilon, \delta)$-differentially private for $1 \leq m \leq K$ and $\delta \geq \Delta \geq 0$. We focus on g being the majority function .

The majority function g is often used in private prediction, where one studies the privacy cost of releasing one prediction [29] and exploits the fact that releasing only the aggregated output on sharded models is significantly more private than releasing each prediction. For example, this occurs in semi-supervised knowledge transfer with private aggregated teacher ensembles (PATE) [86, 87], in ensemble learning algorithms [53, 119], machine unlearning [17], private distributed learning algorithms such as Stochastic Sign-SGD [118], and in ensemble feature selection [68]. Private prediction is also shown to be a competitive technique in data-adaptive settings, where the underlying dataset is changing slowly over time, to quickly adjust to online dataset updates [125]. Furthermore, to address the large privacy loss of private prediction under the many-query regime, there has been recent works in everlasting private prediction that extends privacy guarantees with repeated, possibly infinite, queries without suffering a linear increase in privacy loss [80, 96].

These works, however, rely often on the standard sensitivity analysis of g to provide a private output and thus generally provide limited utility guarantees. This is because the maximum sensitivity of g can be too pessimistic in practice, as observed in the

problem of private hyperparameter optimization [67]. On the other hand, for private model ensembling, a naive way to bound privacy loss without restrictive assumptions is to apply simple composition (Theorem 3.2.2) or general composition (Theorem 3.2.3, a tighter version compared to advanced composition) to reason about the final privacy loss after aggregation. A black-box application of the simple composition theorem to compute g would incur a $K\epsilon$ privacy cost in the pure differential privacy setting, that is, $\delta = 0$, or if one is willing to tolerate some failure probability δ, general composition would yield a $O(\sqrt{K}\epsilon)$ privacy cost [54]. Thus, a natural baseline algorithm $\mathcal{A}$ that is $(m\epsilon, m\Delta)$-differentially private applies privacy amplification by subsampling and randomly chooses m of the K mechanisms to aggregate and returns the majority of the subsampled mechanisms. This technique is reminiscent of the subsampling procedure used for the maximization function g [67] or some general techniques for privacy amplification in the federated setting via shuffling [34].

However, standard composition analysis and privacy amplication techniques can be suboptimal for computing a private majority, in terms of both utility and privacy. Observe that if there is a clear majority among the outputs of $M_1(\mathcal{D}), \ldots, M_K(\mathcal{D})$, one can add less noise. This is because each mechanism M_i is (ϵ, Δ)-differentially private already, and hence, is less likely to change its output on a neighboring dataset by definition. This implies the majority outcome is unlikely to change based on single isolated changes in $\mathcal{D}$. Furthermore, composition theorems make two pessimistic assumptions: 1) the worst-case function g and the dataset $\mathcal{D}$ are considered, and 2) all intermediate mechanism outputs $M_1(\mathcal{D}), \ldots, M_K(\mathcal{D})$ are released, rather than just the final aggregate. Based on these observations, is it possible then to improve the utility of computing a private majority, under a fixed privacy loss?

3.1.1 Our Contributions

We give a (perhaps surprising) affirmative answer to the above question by using our novel data-dependent randomized response framework (DaRRM), which captures all private majority algorithms, we introduce a tractable noise optimization procedure that maximizes the privacy-utility tradeoffs. Furthermore, we can provably achieve a constant factor improvement in utility over simple subsampling by applying data-dependent noise injection when M_i's are i.i.d. and $\delta = 0$. To our knowledge, this is

the first of its work of its kind that gives a tractable utility optimization over the possibly infinite set of privacy constraints.

Data-dependent Randomized Response Majority (DaRRM). We generalize the classical Randomized Response (RR) mechanism and the commonly used subsampling baseline for solving Problem 3.1.1 and propose a general randomized response framework DaRRM (see Algorithm 1), which comes with a customizable noise function γ. We show that DaRRM actually captures all algorithms computing the majority whose outputs are at least as good as a random guess (see Lemma 3.3.3), by choosing different γ functions.

Designing γ with Provable Privacy Amplification. The choice of the γ function in DaRRM allows us to explicitly optimize noise while trading off privacy and utility. Using structural observations, we show privacy amplification by a factor of 2 under mild conditions over applying simple composition in the pure differential privacy setting when the mechanisms M_i's are i.i.d. (see Theorem 3.4.1).

Finding the Best γ through Dimension-Reduced Optimization. We further exploit the generality of DaRRM by applying a novel optimization-based approach that applies constrained optimization to find a data-dependent γ that maximizes some measure of utility. One challenge is that there are infinitely many privacy constraints, which are necessary for DaRRM with the optimized γ to satisfy the given privacy loss. We show that we can reformulate the privacy constraints, which are infinite dimensional, to a finite polynomial-sized constraint set, allowing us to efficiently constrain the optimization problem to find the best γ, even for approximate differential privacy (see Lemma 3.5.1). Empirically, we show that with a small m and ϵ, the optimized γ (see γ_{opt} in Figure 3.2) achieves the best utility among all γ functions, even compared to the subsampling and the data-independent baseline. To our knowledge, this is the first utility maximization algorithm that optimizes over all private algorithms by constrained optimization with dimension reduction.

Experiments. In downstream tasks, such as semi-supervised knowledge transfer for private image classification, we compare our DaRRM with an optimized γ to compute the private label majority from private teachers against PATE [87], which computes the private label majority from non-private teachers. We fix the privacy loss of the output of both algorithms to be the same and find that when the number of teachers K is small, DaRRM indeed has a higher utility than PATE, achieving 10%-

15% and 30% higher accuracy on datasets `MNIST` and `Fashion-MNIST`, respectively.

3.2 Background

3.2.1 Related Work

Private Composition. Blackbox privacy composition analysis often leads to pessimistic utility guarantees. In the blackbox composition setting, one can do no better than the $O(K\epsilon)$ privacy analysis for pure differential privacy [32]. For approximate differential privacy, previous work has found optimal constants for advanced composition by reducing to the binary case of hypothesis testing with randomized response; and optimal tradeoffs between ϵ, δ for black box composition are given in [54], where there could be a modest improvement 20%.

Thus, for specific applications, previous work has turned to white-box composition analysis for improved utility. This includes, for example, moment accountant for private SGD [2] and the application of contractive maps in stochastic convex optimization [37]. For the specific case of model ensembles, [87] shows a data-dependent privacy bound that vanishes as the probability of disagreement goes to 0. Their method provides no utility analysis but they empirically observed less privacy loss when there is greater ensemble agreement.

When g is the maximization function, some previous work shows that an approximately maximum value can be outputted with high probability while incurring $O(\epsilon)$ privacy loss, independently of K. [67] proposed a random stopping mechanism for $m = 1$ that draws samples uniformly at random from $M_i(\mathcal{D})$ at each iteration. In any given iteration, the sampling halts with probability γ and the final output is computed based on the samples collected until that time. This leads to a final privacy cost of only 3ϵ for the maximization function g, which can be improved to 2ϵ [85]. In addition to the aforementioned works, composing top-k and exponential mechanisms also enjoy slightly improved composition analysis via a bounded-range analysis [27, 28].

Bypassing the Global Sensitivity. To ensure differential privacy, it is usually assumed the query function g has bounded global sensitivity — that is, the output of g does not change much on *any* adjacent input datasets differing in one entry. The

noise added to the output is then proportional to the global sensitivity of g. If the sensitivity is large, the output utility will thus be terrible due to a large amount of noises added. However, the worst case global sensitivity can be rare in practice, and this observation has inspired a line of works on designing private algorithms with data-dependent sensitivity bound to reduce the amount of noises added.

Instead of using the maximum global sensitivity of g on any dataset, the classical Propose-Test-Release framework of Dwork [30] uses a local sensitivity value for robust queries that is tested privately and if the sensitivity value is too large, the mechanism is halted before the query release. The halting mechanism incurs some failure probability but deals with the worst-case sensitivity situations, while allowing for lower noise injection in most average-case cases.

One popular way to estimate average-case sensitivity is to use the Subsample-and-Aggregate framework by introducing the notion of *perturbation stability*, also known as *local sensitivity* of a function g on a dataset $\mathcal{D}$ [32, 103], which represents the minimum number of entries in $\mathcal{D}$ needs to be changed to change $g(\mathcal{D})$. One related concept is *smooth sensitivity*, a measure of variability of g in the neighborhood of each dataset instance. To apply the framework under *smooth sensitivity*, one needs to privately estimate a function's local sensitivity L_s and adapt noise injection to be order of $O(\frac{L_s}{\epsilon})$, where L_s can often be as small as $O(e^{-n})$, where $n = |\mathcal{D}|$, the total dataset size [82]. Generally, the private computation of the smooth sensitivity of a blackbox function is nontrivial but is aided by the Subsample and Aggregate approach for certain functions.

These techniques hinge on the observation that a function with higher stability on $\mathcal{D}$ requires less noise to ensure worst case privacy. Such techniques are also applied to answer multiple online functions/queries in model-agnostic learning [13]. However, we highlight two key differences in our setting with a weaker stability assumption. First, in order to estimate the *perturbation stability* of g on $\mathcal{D}$, one needs to downsample or split $\mathcal{D}$ into multiple blocks [13, 32, 103], $\hat{\mathcal{D}}_1, \ldots, \hat{\mathcal{D}}_B$, and estimate the *perturbation stability* based on the mode of $g(\hat{\mathcal{D}}_1), \ldots, g(\hat{\mathcal{D}}_B)$. This essentially reduces the amount of change in the output of g due to a single entry in $\mathcal{D}$, with high probability and replaces the hard-to-estimate *perturbation stability* of g with an easy-to-compute *perturbation stability* of the mode. Such a notion of stability has also been successfully applied, along with the sparse vector technique, for model-agnostic private learning

to handle exponentially number of queries to a model [13]. Note that in these cases, since a private stochastic test is applied, one cannot achieve pure differential privacy [32]. In practice, e.g. federated learning, however, one does not have direct access to $\mathcal{D}$, and thus it is impractical to draw samples from or to split $\mathcal{D}$. Second, to ensure good utility, one relies on a key assumption, i.e. the *subsampling stability* of g, which requires $g(\hat{\mathcal{D}}) = g(\mathcal{D})$ with high probability over the draw of subsamples $\hat{\mathcal{D}}$.

Although our intuition in designing DaRRM also relies on the stability of the mode function g, previous usage of stability to improve privacy-utility tradeoffs, e.g., propose-test-release [32, 106], requires the testing of such stability, based on which one adds a larger (constant) noise γ. This can still lead to adding redundant noise in our case.

Optimal Randomized Response. [49] and [54] show that the classical Randomized Response (RR) mechanism with a constant probability of faithfully revealing the true answer is optimal in certain private estimation problems. Our proposed DaRRM framework and our problem setting is a generalized version of the ones considered in both [49] and [54], which not only subsumes RR but also enables a data-dependent probability, or noise addition.

While RR with a constant probability can be shown optimal in problems such as private count queries or private estimation of trait possession in a population, it is not optimal in other problems, such as private majority ensembling, since unlike the former problems, changing one response of the underlying mechanisms does not necessarily change the output of the majority. To explicitly compute the minimum amout of noise required, one needs the output distributions of the underlying mechanisms but this is unknown. To resolve this, our proposed DaRRM framework adds the amount of noise dependent on the set of observed outcomes from the underlying private mechanisms, $\mathcal{S}$, which is a random variable of the dataset and is hence a proxy. This enables DaRRM to calibrate the amount of noise based on whether the majority output is likely to change. The amount of noise is automatically reduced when the majority output is not likely to change.

Second, [49] and [54] both consider a special case in our setting where all K private mechanisms are i.i.d., while our approach focuses on the more general setting where each private mechanism can have a different output distribution.

Learning A Good Noise Distribution. There have been limited works that

attempt to derive or learn a good noise distribution that improves the utility. For deep neural networks inference, [76] attempts to learn the best noise distribution to maximizing utility subject to an entropy Lagrangian, but no formal privacy guarantees were derived. For queries with bounded sensitivity, [42] demonstrate that the optimal noise distribution is in fact a staircase distribution that approaches the Laplacian distribution as $\epsilon \to 0$.

Private Prediction. Instead of releasing a privately trained model as in private learning, private prediction hides the models and only releases private outputs. Private prediction has been shown as a practical alternative compared to private learning, as performing private prediction is much easier compared to private learning on a wide range of tasks [29, 80, 107]. Although a privately trained model can make infinitely many predictions at the inference time without incurring additional privacy loss, since differential privacy is closed under post-processing, it has been shown recently that it is indeed possible to make infinitely many private predictions [80] with a finite privacy loss for specific problems.

3.2.2 Preliminaries

We first introduce the definition of differential privacy, simple composition and general composition as follows. The general composition [54] gives a near optimal and closed-form bound on privacy loss under adaptive composition, which improves upon advanced composition [32].

Definition 3.2.1 (Differential Privacy (DP) [32])**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with a domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy for $\epsilon, \delta \geq 0$ if for any two* **adjacent datasets** *$\mathcal{D}, \mathcal{D}'$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that $\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta$. $\delta = 0$ is often called pure differential privacy; while $\delta > 0$ is often called approximate differential privacy.*

Theorem 3.2.2 (Simple Composition [32])**.** *For any $\epsilon > 0$ and $\delta \in [0, 1]$, the class of (ϵ, δ)-differentially private mechanisms satisfy $(k\epsilon, k\delta)$-differential privacy under k-fold adaptive composition.*

Theorem 3.2.3 (General Composition (Theorem 3.4 of [54]))**.** *For any $\epsilon > 0, \delta \in [0, 1]$ and $\delta' \in (0, 1]$, the class of (ϵ, δ)-differentially private mechanisms satisfies*

$(\epsilon', 1-(1-\delta)^k(1-\delta'))$-*differential privacy under* k*-fold adaptive composition for*

$$\epsilon' = \min\left\{k\epsilon, \frac{(e^\epsilon-1)\epsilon k}{e^\epsilon+1} + \epsilon\sqrt{2k\log(e+\frac{\sqrt{k\epsilon^2}}{\delta'})}, \frac{(e^\epsilon-1)\epsilon k}{e^\epsilon+1} + \epsilon\sqrt{2k\log(1/\delta')}\right\}$$

We then formalize the error and utility metric in our problem as follows:

Definition 3.2.4 (Error Metric and Utility Metric)**.** *For the problem setting in Definition 3.1.1, let the observed (random) outcomes set be* $\mathcal{S} = \{S_1, .., S_k\}$*, where* $S_i \sim M_i(\mathcal{D})$*. For a fixed* $\mathcal{D}$*, we define the error of an algorithm* $\mathcal{A}$*, i.e.,* $\mathcal{E}(\mathcal{A})$*, in computing the majority function* g *as the Total Variation (TV) distance between* $g(\mathcal{S})$ *and* $\mathcal{A}(\mathcal{D})$*. Specifically,*

$$\mathcal{E}(\mathcal{A}) = \mathcal{D}_{TV}(g(\mathcal{S}) \parallel \mathcal{A}(\mathcal{D})) = |\Pr[\mathcal{A}(\mathcal{D}) = 1] - \Pr[g(\mathcal{S}) = 1]|$$

and the utility is defined as $1 - \mathcal{E}(\mathcal{A})$*.*

Notation. Throughout the paper, we use the same notations defined in Problem 3.1.1 and Definition 3.2.4. Furthermore, let $\mathcal{D}$ and $\mathcal{D}'$ to denote a pair of adjacent datasets with one entry being different. Also, let $p_i = \Pr[M_i(\mathcal{D}) = 1]$ and $p_i' = \Pr[M_i(\mathcal{D}') = 1]$, $\forall i \in [K]$. We omit the subscript i when all p_i's or p_i''s are equal. $\mathbb{I}\{\cdot\}$ denotes the indicator function and $[K] = \{1, 2, \ldots, K\}$. For the purpose of analysis, let $\mathcal{L}(\mathcal{D}) = \sum_{i=1}^K M_i(\mathcal{D}) \in \{0, 1, \ldots, K\}$, i.e. the (random) sum of all observed outcomes on dataset $\mathcal{D}$. $\mathcal{D}$ is omitted when the context is clear. Unless specified, we use the noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ as input to our algorithms to calibrate the probabilistic noise injection. Unless specified, the privacy allowance $m \in \mathbb{R}$.

3.3 Private Majority Algorithms

The very first approach to consider when solving private majority ensembling (Problem 3.1.1), since the output is binary, is the classical Randomized Response (RR) mechanism [32], where one flips a biased coin with a *constant* probability $p_{const} \in [0, 1]$. If the coin lands on head with probability p_{const}, output the true majority base on K samples; if not, then simply output a noisy random answer. However, to make the

output $(m\epsilon, \delta)$-differential private, the success probability p_{const} can be at most $O(\frac{m}{K})$ (or $O(\frac{m}{\sqrt{K}})$) when $\delta = 0$ (or $\delta > 0$) (see Appendix B.1.1), which is too small for any reasonable utility.

The key observation for improved utility is that the probability of success should not be a *constant*, but should depend on the *unpublished* set of observed outcomes from the mechanisms $\mathcal{S}$. If we see many 1's or 0's in $\mathcal{S}$, then there should be a clear majority even on adjacent datasets. On the other hand, if we see about half 1's and half 0's, this means the majority is highly volatile to data changes, which implies we need more noise to ensure privacy. In summary, if we can calibrate the success probability based on $\mathcal{S}$ to smoothly increase when there is a clear majority, we can improve the utility without affecting privacy.

Subsampling. One natural baseline is outputting the majority of m out of K randomly subsampled mechanisms (without replacement), given a privacy allowance $m \in [K]$. Suppose $\delta \geq m\Delta$, the privacy loss of the aggregated output can be reasoned through simple composition or general composition. Interestingly, we show outputting the majority of m out of K subsampled mechanisms corresponds to RR with a *non-constant* probability $p_\gamma = \gamma_{Sub}(\mathcal{L}(\mathcal{D}))$, which is set by a polynomial function $\gamma_{Sub} : \{0, \ldots, K\} \to [0, 1]$ based on the sum of observed outcomes $\mathcal{L}(\mathcal{D})$ in Lemma 3.3.1 (see a full proof in Appendix B.1.2). Intuitively, subsampling may be seen as implicitly adding noise by only outputting based on a randomly chosen subset of the mechanisms; therefore this implicit noise is inherently *data-dependent* on $\mathcal{L}(\mathcal{D})$.

Lemma 3.3.1. *Consider Problem 3.1.1, with the privacy allowance $m \in [K]$. Consider the data-dependent algorithm that computes $\mathcal{L}(\mathcal{D})$ and then applies RR with probability p_γ. If $p_\gamma = \gamma_{Sub}(l)$, where $l \in \{0, 1, \ldots, K\}$ is the value of $\mathcal{L}(\mathcal{D})$, i.e., the (random) sum of observed outcomes on dataset $\mathcal{D}$, and $\gamma_{Sub} : \{0, 1, \ldots, K\} \to [0, 1]$ is*

$$\gamma_{Sub}(l) = \gamma_{Sub}(K - l) = \begin{cases} 1 - 2\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} & \text{if } m \text{ is odd} \\ 1 - 2\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} & \text{if } m \text{ is even} \end{cases}$$

then the majority of m out of K subsampled mechanisms without replacement and the output of our data-dependent RR algorithm have the same distribution.

One thing special about subsampling is that when $m = 1$, it indeed results in

Algorithm 1 DaRRM$(\cdot)$: Data-dependent Randomized Response Majority

Input: K (ϵ, Δ)-DP mechanisms $\{M_i\}_{i=1}^K$, noise function $\gamma : \{0,1\}^{K+1} \to [0,1]$ (in our specific setting $\gamma : \{0, 1, \ldots, K\} \to [0,1]$), dataset $\mathcal{D}$, privacy allowance $1 \leq m \leq K$, failure probability $\delta \geq \Delta \geq 0$
Output: $(m\epsilon, \delta)$-DP majority vote of $\{M_i\}_{i=1}^K$
$\mathcal{S} = \{S_1, .., S_K\}$, where $S_i \sim M_i(\mathcal{D})$
$\mathcal{L} = \sum_{i=1}^K S_i$
Set probability $p_\gamma \leftarrow \gamma(\mathcal{S})$ (in our setting $p_\gamma \leftarrow \gamma(\mathcal{L})$)
Flip the p_γ- biased coin
if Head (with probability p_γ) **then**
    Output $\mathbb{I}\{\frac{1}{K}\mathcal{L} \geq \frac{1}{2}\}$
else
    Output 0/1 with equal probability
end if

the optimal error, which we show in Lemma 3.3.2 as follows. See a full proof in Appendix B.1.3. Note that when $m = 1$, subsampling outputs a majority of 1 with probability exactly $\frac{1}{K}\sum_{i=1}^K p_i$.

Lemma 3.3.2 (Lower Bound on Error when $m = 1$)**.** *Let $\mathcal{A}$ be an (ϵ, δ)-differentially private algorithm, where $0 \leq \epsilon < c$ for some constant $c > 0$ and $\delta \in [0, \frac{1}{2})$, that computes the majority of K (ϵ, δ)-differentially private mechanisms $M_1, \ldots, M_K$, where $M_i : \mathcal{D} \to \{0,1\}$ on dataset $\mathcal{D}$ and $\Pr[M_i(\mathcal{D}) = 1] = p_i, \forall i \in [K]$. Then, the error $\mathcal{E}(\mathcal{A}) \geq |\Pr[g(\mathcal{S}) = 1] - \frac{1}{K}\sum_{i=1}^K p_i|$, where $g(\mathcal{S})$ is the probability of the true majority output being 1 as defined in Definition 3.1.1.*

Data-dependent Randomized Response (DaRRM). Does subsampling give optimal utility when $m > 1$? Inspired by the connection between RR and subsampling, we propose Data-dependent Randomized Response Majority (DaRRM) in Algorithm 1, to study optimizing privacy-utility tradeoffs in private majority ensembling. In particular, DaRRM has a *non-constant* success probability p_γ that is set by a parameterized noise function γ, which in turn depends on the set of observed outcomes $\mathcal{S} = \{S_1, \ldots, S_K\}$. In fact, we can show that DaRRM is general: any *reasonable* algorithm $\mathcal{A}$, name one whose output is at least as good as a random guess, can be captured by the DaRRM framework in Lemma 3.3.3 (see a full proof in Appendix B.1.4). We denote DaRRM instantiated with a specific noise function γ by DaRRM_γ.

Lemma 3.3.3 (Generality of DaRRM)**.** *Let $\mathcal{A}$ be any randomized algorithm to compute the majority function g on $\mathcal{S}$ such that for all $\mathcal{S}$, $\Pr[\mathcal{A}(\mathcal{S}) = g(\mathcal{S})] \geq 1/2$ (i.e. $\mathcal{A}$ is at least as good as a random guess). Then, there exists a a general function $\gamma : \{0,1\}^{K+1} \to [0,1]$ such that if one sets p_γ by $\gamma(\mathcal{S})$ in DaRRM, the output distribution of DaRRM$_\gamma$ is the same as the output distribution of $\mathcal{A}$.*

Designing the γ Function. With the DaRRM framework, we ask: how to design a good γ function that maximizes the utility? First, we introduce two characteristics of γ that do not affect the utility, while simplifying the analysis and the empirical optimization:

(a) **A function of the sum of observed samples**: Since the observed samples set $\mathcal{S}$ is a permutation-invariant set, a sufficient statistic that captures the full state of $\mathcal{S}$ is $\mathcal{L} = \sum_{i=1}^{K} S_i$, the sum of observed outcomes. This allows us to reduce $\gamma(\mathcal{S}) = \gamma(\mathcal{L})$. *Hence, in the rest of the paper, we focus on* $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$.

(b) **Symmetric around $\frac{K}{2}$**: If γ is asymmetric, we can symmetrize by reflecting one region about $\frac{K}{2}$ and achieve better or equal expected utility, where the utility is summed over symmetric distributions of p_i.

Note that γ_{Sub} satisfies both characteristics. Now, recall $\mathcal{L}(\mathcal{D})$ and $\mathcal{L}(\mathcal{D}')$ are the sum of observed outcomes on adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$. Also, recall $p_i = \Pr[M_i(\mathcal{D}) = 1]$ and $p'_i = \Pr[M_i(\mathcal{D}') = 1]$ are the output probabilities of the mechanism M_i on $\mathcal{D}, \mathcal{D}'$. To design a good noise function γ in DaRRM, we start by deriving conditions for a γ function such that DaRRM$_\gamma$ is $(m\epsilon, \delta)$-differentially private in Lemma 3.3.4 (see a full proof in Appendix B.1.5).

Lemma 3.3.4 (γ privacy condition)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, let $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, where $\mathcal{D}$ and $\mathcal{D}'$ are adjacent datasets and $l \in \{0, \ldots, K\}$. For a noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ such that $\gamma(l) = \gamma(K - l), \forall l$, DaRRM$_\gamma$ is $(m\epsilon, \delta)$-differentially private if and only if for all α_l, α'_l, the following holds,*

$$f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma) \leq e^{m\epsilon} - 1 + 2\delta \tag{3.1}$$

where f is called the ***privacy cost objective*** *and*

$$f(p_1,\ldots,p_K,p'_1,\ldots,p'_K;\gamma) := \sum_{l=0}^{\frac{K-1}{2}}(e^{m\epsilon}\alpha'_l-\alpha_l)\cdot\gamma(l)+\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l-e^{m\epsilon}\alpha'_l)\cdot\gamma(l)$$

3.4 Provable Privacy Amplification

We theoretically demonstrate that privacy is provably amplified under improved design of γ in our DaRRM framework. Specifically, we show when the mechanisms are i.i.d. and $\delta = 0$, we gain privacy amplification by a factor of 2 compared to the naïve subsampling baseline by carefully designing γ.

Theorem 3.4.1 (Provable Privacy Amplification by 2)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, with i.i.d. mechanisms $\{M_i\}_{i=1}^K$, i.e., $p_i = p$, $p'_i = p'$, $\forall i \in [K]$, the privacy allowance $m \in [K]$ and $\delta = \Delta = 0$. Let the noise function $\gamma : \{0,1,\ldots,K\} \to [0,1]$ be that:*
if $m \geq \frac{K+1}{2}$, $\gamma(l) = 1$ and if $m \leq \frac{K-1}{2}$,

$$\gamma(l) = \begin{cases} 1-2h(l) & \forall l \leq \frac{K-1}{2} \\ 2h(l)-1 & \forall l \geq \frac{K+1}{2} \end{cases}$$

where $h(l) = \sum_{i=m}^{2m-1} \frac{\binom{l}{i}\binom{K-l}{2m-1-i}}{\binom{K}{2m-1}}$, then DaRRM$_\gamma$ is $m\epsilon$-differentially private.

Interpretation. First, when $m \leq \frac{K-1}{2}$ is small, the $\gamma(l)$ in Theorem 3.4.1 corresponds to outputting the majority based on subsampling $2m-1$ outcomes, from Lemma 3.3.1. However, the subsampling baseline, whose privacy loss is reasoned through simple composition, would have indicated that one can only output the majority based on m outcomes, therefore implying a 2x privacy gain. When $m \geq \frac{K+1}{2}$, the above theorem indicates that we can set a constant $\gamma = 1$, which implies we are optimally outputting the true majority with no noise while still surprisingly ensuring $m\epsilon$ privacy.

Intuition. This 2x privacy gain is intuitively possible because the majority is only dependent on half of the mechanisms' outputs, therefore the privacy leakage is also halved. To see this, we start by analyzing the privacy cost objective in Eq. B.28,

where with a careful analysis of its gradient, we show that the maximum indeed occurs $(p^*, p'^*) = (0, 0)$ when γ satisfies certain conditions. Now, when $(p^*, p'^*) \to 0$, note that the probability ratio of outputting 1 with $2m-1$ outcomes is approximately $e^{m\epsilon}$, where dependence on m follows because the probability of outputting 1 is dominated by the probability that exactly m mechanisms output 1. To rigorize this, we derive sufficient conditions for γ functions that satisfy $\max_{(p,p')} f(p, p'; \gamma) = f(0, 0; \gamma) \leq e^{m\epsilon} - 1$ as indicated by Lemma 3.3.4, to ensure DaRRM to be $m\epsilon$-differentially private and a more detailed overview and the full proof can be found in Appendix B.2.

3.5 Optimizing the Noise Function γ in DaRRM

Theoretically designing γ and extending privacy amplification results to the $\delta > 0$ case is difficult and it is likely that our crafted γ is far from optimal. On the other hand, one can optimize for such γ^* that maximizes the utility but this involves solving a "Semi-infinite Programming" problem, due to the infinitely many privacy constraints, which are the constraints in the optimization problem necessary to ensure DaRRM with the optimized γ satisfy a given privacy loss. Solving a "Semi-infinite Programming" problem in general is non-tractable, but we show that in our specific setting this is in fact tractable, proposing a novel learning approach based on DaRRM that can optimize the noise function to maximize the utility. To the best of our knowledge, such optimization, presented as follows, is the first of its kind:

$$\min_{\gamma \in [0,1]^{K+1}} \mathbb{E}_{p_1, p_2, \ldots, p_K \sim \mathcal{T}}[\mathcal{E}(\mathsf{DaRRM}_\gamma)] \tag{3.2}$$

$$\text{s.t.} \max_{\{(p_i, p_i') \in \mathcal{F}_i\}_{i=1}^K} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma) \leq e^{m\epsilon} - 1 + 2\delta \tag{3.3}$$

$$\gamma(l) = \gamma(K - l), \forall l \in \{0, 1, \ldots, K\}$$

where f is the privacy cost objective as defined in Lemma 3.3.4, $\mathcal{F}_i$ is the feasible region where (p_i, p_i') lies due to each mechanism M_i being ϵ-differentially private. Observe that since γ is symmetric around $\frac{K}{2}$, we only need to optimize $\frac{K+1}{2}$ variables instead of $K+1$ variables. $\mathcal{T}$ is the distribution from which $p_1, \ldots, p_K$ are drawn. We want to stress that no prior knowledge about the dataset or the amount of consensus among the private mechanisms is required to use our optimization framework. When

there is no prior knowledge about $p_1, \ldots, p_K$, $\mathcal{T}$ is set to be the uniform distribution for maximizing the expected utility. Note the above optimization problem also enables the flexibility of incorporating prior knowledge about the mechanisms by choosing a prior distribution $\mathcal{T}$ to further improve the utility.

Optimizing Over All Algorithms. We want to stress that by solving the above optimization problem, we are indeed optimizing over *all* algorithms for maximal utility, since we show in Lemma 3.3.3 DaRRM that captures *all reasonable* algorithms computing a private majority.

Linear Optimization Objective. Perhaps surprisingly, it turns out that optimizing for γ^* is a Linear Programming (LP) problem! Indeed, after expanding the optimization objective in Eq. 3.2 by the utility definition (see Definition 3.2.4), optimizing the above objective is essentially same as optimizing:

$$\min_{\gamma \in [0,1]^{K+1}} -\frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} \mathbb{E}_{p_1,p_2,\ldots,p_K \sim \mathcal{T}} \left[(\alpha_l - \alpha_{K-l}) \right] \cdot \gamma(l)$$

where $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l], \forall l \in \{0, 1, \ldots, K\}$ and observe $\mathcal{L}(\mathcal{D}) \sim \text{PoissonBinomial}(p_1, \ldots, p_K)$. The above objective is linear in γ. See a full derivation in Appendix B.3.1.

Although taking the expectation over $p_1, \ldots, p_K$ involves integrating over K variables and this can be computationally expensive, we discuss how to formulate a computationally efficient approximation of the objective in Appendix B.3.2, which we later use in the experiments. Note that the objective only for maximizing the utility and hence approximating the objective does not affect the privacy guarantee.

Reducing Infinitely Many Constraints to A Polynomial Set. The constraints in the optimization problem (Eq. 3.3) is what makes sure the output of DaRRM_γ is $m\epsilon$-differentially private. We thus call them *the privacy constraints*. Note that the privacy constraints are linear in γ.

Though it appears we need to solve for infinitely many such privacy constraints since p_i's and p_i''s are continuous, we show that through a structural understanding of DaRRM, we can reduce the number of privacy constraints from infinitely many to exponentially many, and further to a polynomial set. First, we observe the privacy cost objective f is linear in each independent pair of (p_i, p_i') fixing all (p_j, p_j'), $\forall j \neq i$, and hence finding the worst case probabilities in (p_i, p_i') given any

γ, $(p_i^*, p_i'^*) = \text{argmax}_{(p_i, p_i')} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$ is a linear programming (LP) problem. Furthermore, since p_i and p_i' are the probability of outputting 1 from the i-th (ϵ, Δ)-differentially private mechanism M_i on adjacent datasets, by definition, they are close and lie in a feasible region $\mathcal{F}_i$, which we show has 8 corners if $\delta > 0$ (and only 4 corners if $\delta = 0$). This implies $(p_i^*, p_i'^*)$ only happens at one of the corners of $\mathcal{F}_i$, and hence the number of constraints reduces to K^8 (and K^4 if $\delta = 0$). Second, observe that α_l and α_l' in the privacy cost objective f are the pmf of two Poisson Binomial distributions at $l \in \{0, \ldots, K\}$. Notice that the Poisson Binomial is invariant under the permutation of its parameters, i.e. PoissonBinomial$(p_1, \ldots, p_K)$ has the same distribution as PoissonBinomial$(\pi(p_1, \ldots, p_K))$, under some permutation π. Based on this observation, we show the number of constraints can be further reduced to $O(K^7)$ if $\delta > 0$ (and $O(K^3)$ if $\delta = 0$). We formalize the two-step reduction of the number of privacy constraints in Lemma 3.5.1 as follows. See a full proof in Appendix B.3.3. [1]

Lemma 3.5.1. *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1 and let f be the privacy cost objective as defined in Lemma 3.3.4. Given an arbitrary noise function γ, let the worst case probabilities be $(p_1^*, \ldots, p_K^*, p_1'^*, \ldots, p_K'^*) = \text{argmax}_{\{(p_i, p_i')\}_{i=1}^K} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$.*

$$(p_1^*, \ldots, p_K^*, p_1'^*, \ldots, p_K'^*) = \underset{\{(p_i, p_i')\}_{i=1}^K}{\text{argmax}} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$$

Then, each pair $(p_i^*, p_i'^*), \forall i \in [K]$ satisfies

$$(p_i^*, p_i'^*) \in \{(0,0), (1,1), (0,\Delta), (\Delta, 0), (1-\Delta, 1), (1, 1-\Delta), (\frac{e^\epsilon + \Delta}{e^\epsilon + 1}, \frac{1-\Delta}{e^\epsilon + 1}), (\frac{1-\Delta}{e^\epsilon + 1}, \frac{e^\epsilon + \Delta}{e^\epsilon + 1})\}$$

Furthermore, when $\delta > 0$, there exists a finite vector set $\mathcal{P}$ of size $O(K^7)$ such that if $\beta = \max_{\{(p_i, p_i')\}_{i=1}^K \in \mathcal{P}} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$, then $f(p_1^*, \ldots, p_K^*, p_1'^*, \ldots, p_K'^*; \gamma) \leq \beta$. When $\delta = 0$, the size of $\mathcal{P}$ can be reduced to $O(K^3)$.

[1] **Practical Limitation.** Although the number of constraints is polynomial in K and optimizing γ in DaRRM is an LP, $O(K^7)$ can still make the number of constraints intractably large when K is large. In practice, we observe with the `Gurobi` optimizer, one can optimize γ for $K \leq 41$ on a laptop if $\delta > 0$. But if $\delta = 0$, since the number of privacy constraints is $O(K^3)$, one can optimize for K over 100.

3.6 Experiments

We empirically solve[2] the above optimization problem (Eq. 3.2) using the `Gurobi`[3] solver and first present the shape of the optimized γ function, which we call γ_{opt}, and its utility in Section 3.6.1. Then, we demonstrate the compelling effectiveness of DaRRM with an optimized γ function, i.e., $\mathsf{DaRRM}_{\gamma_{opt}}$, in ensembling labels for private prediction from private teachers through the application of semi-supervised knowledge transfer for private image classification in Section 3.6.2.

3.6.1 Optimized γ in Simulations

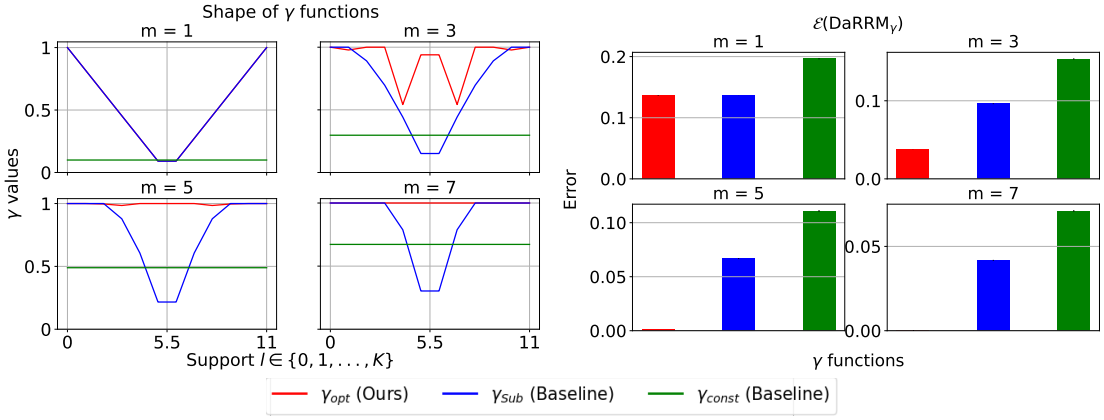

Figure 3.2: Plots of the shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: the optimized γ_{opt}, and the baselines γ_{Sub} (corresponding to subsampling) and γ_{const} (corresponding to RR). Here, $K = 11, m \in \{1, 3, 5, 7\}$, $\epsilon = 0.1$, $\Delta = 10^{-5}$ and $\delta = 1-(1-\Delta)^m \approx m\Delta$.

We compare the shape and the error $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: an optimized γ_{opt} and the subsampling γ_{Sub} as in Lemma 3.3.1[4]. We also compare against p_{const} in the classical baseline RR (see Section B.1.1) and $\mathcal{E}(\mathsf{RR})$. Here, p_{const} can be

[2]All code for the experiments can be found at `https://anonymous.4open.science/r/OptimizedPrivateMajority-CF50`

[3]`https://www.gurobi.com/`

[4]Note the subsampling mechanism from Section 3.4, which enjoys a privacy amplification by a factor of 2, only applies to pure differential privacy settings (i.e., when $\Delta = \delta = 0$). However, we focus on the more general approximate differential privacy settings (with $\Delta > 0$) in the experiments, and hence, the subsampling baseline we consider throughout this section is the basic version without privacy amplification. To see how the subsampling mechanism from Section 3.4 with privacy amplification compares against the other algorithms, please refer to Appendix B.4.1.

viewed as a constant noise function $\gamma_{const}(l) = p_{const}, \forall l \in \{0, 1, \ldots, K\}$; and $\mathcal{E}(\mathsf{RR})$ is the same as $\mathcal{E}(\mathsf{DaRRM}_{\gamma_{const}})$.

We present the results with $K = 11, \epsilon = 0.1, \Delta = 10^{-5}$ and $m \in \{1, 3, 5, 7\}$. We assume there is no prior knowledge about the mechanisms $\{M_i\}_{i=1}^K$, and set the prior distribution from which p_i's are drawn, $\mathcal{T}$, to be the uniform distribution, in the optimization objective (Eq. 3.2) searching for γ_{opt}. To ensure a fair comparison against the subsampling baseline, we set δ to be the one by m-fold general composition (see Theorem 3.2.3), which in this case, is $\delta = 1 - (1 - \Delta)^m \approx m\Delta$. We plot each γ functions over the support $\{0, 1, \ldots, K\}$ and the corresponding error of each algorithm in Figure 3.2.

Discussion. In summary, at $m = 1$, the optimized noise function γ_{opt} overlaps with γ_{sub} which corresponds to the subsampling baseline. This agrees with our lower bound on the error in Lemma 3.3.2, which implies that at $m = 1$, subsampling indeed gives the optimal error. When $m > 1$, the optimized noise function γ_{opt} has the highest probability of outputting the true majority over the support than the γ functions corresponding to the baselines. This implies $\mathsf{DaRRM}_{\gamma_{opt}}$ has the lowest error (and hence, highest utility), which is verified on the bottom set of plots. More results on comparing the $\mathsf{DaRRM}_{\gamma_{opt}}$ optimized under the uniform $\mathcal{T}$ against the baselines by general composition (Theorem 3.2.3) and in pure differential privacy settings (i.e., $\Delta = \delta = 0$) for large K and m can be found in Appendix B.4.1 and B.4.1. Furthermore, we include results optimizing γ using a non-uniform $\mathcal{T}$ prior in Appendix B.4.1.

3.6.2 Private Semi-Supervised Knowledge Transfer

Semi-supervised Knowledge Transfer. We apply our DaRRM framework in the application of semi-supervised knowledge transfer for private image classification. We follow a similar setup as in PATE [86, 87], where one trains K teachers, each on a subset of a sensitive dataset, and at the inference time, queries the teachers for the majority of their votes, i.e., the predicted labels, of a test sample. Each time the teachers are queried, there is a privacy loss, and we focus on this private prediction subroutine in this section. To limit the total privacy loss over all queries, the student model is also trained on a public dataset without labels. The student model queries the labels of a small portion of the samples in this dataset from the teachers and

Dataset	MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.63 (0.09)	0.76 (0.09)	**0.79 (0.09)**
$Q = 50$	0.66 (0.06)	0.75 (0.06)	**0.79 (0.05)**
$Q = 100$	0.64 (0.04)	0.76 (0.04)	**0.80 (0.04)**

Dataset	Fashion-MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.65 (0.11)	0.90 (0.07)	**0.96 (0.03)**
$Q = 50$	0.59 (0.06)	0.94 (0.03)	**0.96 (0.02)**
$Q = 100$	0.64 (0.04)	0.93 (0.02)	**0.96 (0.02)**

Table 3.1: Accuracy of the predicted labels of Q query samples on datasets MNIST (on the left) and Fashion-MNIST (on the right). We report the mean and one std. in parentheses over 10 random draws of the query samples from the test dataset. Note each prediction on the query sample is $(\epsilon_{query}, \delta_{query})$-differentially private. With the same per query privacy loss (and hence the same total privacy loss over Q samples), DaRRM$_{\gamma_{opt}}$ achieves the highest accuracy compared to the other two baselines.

is then trained using semi-supervised learning algorithms on both the labeled and unlabeled samples from the public dataset.

Baselines. We want the privacy loss per query of a test sample to the teachers to be $(\epsilon_{query}, \delta_{query})$. This can be achieved via two ways: 1) Train K non-private teachers, add Gaussian noise to the number of predicted labels from the teachers in each output class, and output the majority of the noisy votes. This is exactly the GNMax algorithm from PATE [87]. 2) Train K (ϵ, Δ)-differentially private teachers and output the majority of the teachers' votes by adding a smaller amount of noise. This can be computed using DaRRM with an appropriate noise function γ. We compare the performance of GNMax and DaRRM with two γ functions: γ_{opt} (i.e., the optimized γ), and γ_{Sub} (i.e., the subsampling baseline). The overall privacy loss over Q queries to the teachers can be computed by general composition (Theorem 3.2.3).

Experiment Setup. We use samples from two randomly chosen classes — class 5 and 8 — from the MNIST and Fashion-MNIST datasets to form our training and testing datasets. Our MNIST has a total of 11272 training samples and 1866 testing samples; our Fashion-MNIST has 10000 training samples and 2000 testing samples. We train $K = 11$ teachers on equally divided subsets of the training datasets. Each teacher is a CNN model. The non-private and private teachers are trained using SGD and DP-SGD [2], respectively, for 5 epochs. *DaRRM Setup:* The Gaussian noise in DP-SGD has zero mean and std. $\sigma_{dpsgd} = 12$; the gradient norm clipping threshold is $C = 1$. This results in each private teacher, trained on MNIST and Fashion-MNIST, being $(\epsilon, \Delta) = (0.0892, 10^{-4})$ and $(0.0852, 10^{-4})$-differentially private, respectively,

after 5 epochs. We set the privacy allowance $m = 3$[5] and the privacy loss per query is then computed using general composition under m-fold, which give the same privacy loss in the high privacy regime, resulting in $(\epsilon_{query}, \delta_{query}) = (0.2676, 0.0003)$ on `MNIST` and $(0.2556, 0.0003)$ on `Fashion-MNIST`. *GNMax Setup:* We now compute the std. σ of the Gaussian noise used by GNMax to achieve a per-query privacy loss of $(m\epsilon, m\Delta)$, as in the DaRRM setup. We optimize σ according to the Renyi differential privacy loss bound of Gaussian noise. Although [87] gives a potentially tighter data-dependent privacy loss bound for majority ensembling *non-private* teachers, we found when K and the number of output classes are small as in our case, even if all teachers agree on a single output class, the condition of the data-dependent bound is not satisfied. Hence, we only use the privacy loss bound of Gaussian noise here to set σ in GNMax. See Appendix B.4.2 for more details, including the σ values and other parameters. Finally, the per sample privacy loss and the total privacy loss over Q queries, which is computed by advanced composition, are reported in Table B.7.

The testing dataset is treated as the public dataset on which one trains a student model. [87] empirically shows querying $Q = 1\%N$ samples from a public dataset of size N suffices to train a student model with a good performance. Therefore, we pick $Q \in \{20, 50, 100\}$. We repeat the selection of Q samples 10 times and report the mean test accuracy with one std. in parentheses in Table 3.1. The Q queries serve as the labeled samples in training the student model. The higher the accuracy of the labels from the queries, the better the final performance of the student model. We skip the actual training of the student model using semi-supervised learning algorithms here.

Discussion. Table 3.1 shows $\text{DaRRM}_{\gamma_{opt}}$ achieves the highest accuracy (i.e., utility) compared to the two baselines on both datasets. First, comparing to $\text{DaRRM}_{\gamma_{Sub}}$, we verify that subsampling does not achieve a tight privacy-utility tradeoff, and we can optimize the noise function γ in DaRRM to maximize the utility given a target privacy loss. Second, comparing to GNMax, the result shows there are regimes

[5]Here, we present results with privacy allowance $m = 3$ because we think this is a more interesting case. $m = 1$ is less interesting, since one cannot get improvement compared to the subsampling baseline. m close to a $\frac{K}{2} \approx 5$ is also less interesting, as this case seems too easy for our proposed method (the optimized γ function is very close to 1, meaning very little noise needs to be added in this case). Hence, we pick $m = 3$, which is a case when improvement is possible, and is also potentially challenging for our optimization framework. This is also realistic as most applications would only want to tolerate a constant privacy overhead. See more results with different privacy allowance m's in this setting in Appendix B.4.2.

Dataset	# Queries	Privacy loss per query $(\epsilon_{query}, \delta_{query})$	Total privacy loss over Q queries $(\epsilon_{total}, \delta_{total})$
MNIST	$Q = 20$	$(0.2676, 0.0003)$	$(5.352, 0.006)$
	$Q = 50$		$(9.901, 0.015)$
	$Q = 100$		$(15.044, 0.030)$
Fashion MNIST	$Q = 20$	$(0.2556, 0.0003)$	$(5.112, 0.006)$
	$Q = 50$		$(9.382, 0.015)$
	$Q = 100$		$(14.219, 0.030)$

Table 3.2: The privacy loss per query to the teachers and the total privacy loss over Q queries. Note the total privacy loss is computed by general composition (see Theorem 3.2.3), where we set $\delta' = 0.0001$.

where ensembling private teachers gives a higher utility than directly ensembling non-private teachers, assuming the outputs in both settings have the same privacy loss. Intuitively, this is because ensembling private teachers adds fine-grained noise during both training the teachers and aggregation of teachers' votes, while ensembling non-private teachers adds a coarser amount of noise only to the teachers' outputs. This further motivates private prediction from private teachers and the practical usage of DaRRM, in addition to the need of aggregating private teachers in federated learning settings with an honest-but-curious server.

3.7 Conclusion

In computing a private majority from K private mechanisms, we propose the DaRRM framework, which is provably general, with a customizable γ function. We show a privacy amplification by a factor of 2 in the i.i.d. mechanisms and a pure differential privacy setting. For the general setting, we propose an tractable optimization algorithm that maximizes utility while ensuring privacy guarantees. Furthermore, we demonstrate the empirical effectiveness of DaRRM with an optimized γ. We hope that this work inspires more research on the intersection of privacy frameworks and optimization.

Chapter 4

Private Optimization: Private Incremental Gradient (IG) Methods with Public Data

This chapter is based on an ongoing work.

In this chapter, we focus on private optimization algorithms, which is the cornerstone of privacy preserving machine learning. Specifically, we study differentially private optimization for empirical risk minimization, focusing on shuffled gradient methods, which, unlike stochastic gradient descent (SGD), do not rely on i.i.d. sampling, offering broader practical applicability. While differentially private SGD (DPSGD) benefits from well-established privacy amplification via subsampling, private shuffled gradient methods lack this advantage and remain underexplored. Privacy amplification by iteration (PABI) offers a promising alternative for private shuffled gradient methods, especially when public data is available. However, a unified analysis of privacy and convergence trade-offs for these methods is still lacking.

In this work, we examine Incremental Gradient (IG) methods – a basic form of shuffled gradient methods – for private optimization over bounded domains with convex, smooth, and Lipschitz objectives. To analyze differentially private IG (DP-IG) and leverage public data for optimal privacy-performance trade-offs, we develop a generalized theoretical framework with the flexibility of noise injection and using

surrogate objectives.

Within this framework, we introduce a novel dissimilarity metric specifically suited to measure the gap between the true objective and surrogate objectives in shuffled gradient methods. Based on this framework, we show the first empirical excess risk bound for DP-IG, $O(\frac{1}{n^{2/3}}(\frac{\sqrt{d}}{\epsilon})^{4/3})$, where n is the number samples, d the dimension, and ϵ is the privacy loss. We also propose an interleaved training method using public and private samples, further reducing empirical excess risk. Finally, we empirically validate the effectiveness of DP-IG with interleaved training against several baselines across various tasks utilizing public data.

4.1 Introduction

Differential privacy (DP) has emerged as a cornerstone of privacy-preserving machine learning, offering robust guarantees against the leakage of sensitive information in training datasets. In this work, we focus on solving empirical risk minimization (ERM) problems under differential privacy constraints. Given a training dataset $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$, where $\mathbf{d}_i \in \mathbb{R}^d, \forall i \in [n]$, the goal is to minimize the following composite objective function:

$$\min_{\mathbf{x}\in\mathcal{Q}} F(\mathbf{x}) + \psi(\mathbf{x}), \text{ where } F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \mathbf{d}_i) := f_i(\mathbf{x})$$

where ψ is a regularization function, $\mathcal{Q}$ is a bounded convex constraint set and $f_i(\mathbf{x})$ is assumed to be convex, smooth and Lipschitz[1]. Note in such a case, the bounded domain $\mathcal{Q}$ is required, or this leads to a vacuous set of objective functions otherwise.

The standard and widely used approach for differentially private optimization in this setting is Differentially Private Stochastic Gradient Descent (DP-SGD), also known as Noisy-SGD. DP-SGD builds upon the classical Stochastic Gradient Descent (SGD) algorithm by adding noise to gradient updates to ensure differential privacy. However, both SGD and DP-SGD rely on independent and identically distributed (i.i.d.) sampling of data with replacement at each step. While effective in theory, this

[1]Note that convexity and smoothness are standard assumptions in optimization literature. Lipschitzness is necessary only for privacy analysis, and this assumption can be removed by using gradient clipping.

approach faces practical challenges, including non-fixed batch sizes and scalability issues [22, 23]. Many real-world optimization algorithms, both private and non-private, instead adopt shuffled gradient methods, where the order of samples used for gradient computation is determined before the start of each epoch. Unlike i.i.d. sampling with replacement, shuffled gradient methods use sampling without replacement, which can be categorized into three subtypes:

1. **Incremental Gradient (IG) Methods**: The order of samples is fixed at the start of training.
2. **Shuffle Once (SO)**: Samples are shuffled once before training begins.
3. **Random Reshuffling (RR)**: Samples are reshuffled at the beginning of each epoch.

Empirically, shuffled gradient methods have been observed to converge faster than SGD in non-private settings due to their lower variance in gradient computation. This faster convergence has been theoretically proven in recent works, showing that shuffled gradient methods achieve a convergence rate of $\mathcal{O}((\frac{1}{K})^{2/3})$ [69], where K is the number of epochs, compared to the $O(\frac{1}{\sqrt{T}})$ rate of SGD, where $T = nK$ is the total number of gradient steps.

Incorporating differential privacy into shuffled gradient methods raises unique challenges. While DP-SGD leverages privacy amplification by subsampling to achieve better privacy-utility trade-offs, this amplification technique is unavailable in shuffled gradient methods due to the predetermined order of sample usage. An alternative mechanism, known as privacy amplification by iteration (PABI), has been explored [7, 69, 121]. PABI assumes that intermediate model parameters are not released during training, providing a natural privacy amplification by only exposing the final model parameters. However, current studies on PABI focus primarily on privacy analysis, without exploring its impact on convergence rates. Moreover, in practice, private shuffled gradient methods often use the same noise variance as DP-SGD [22, 23], without considering whether this is correct or its impact on convergence in shuffled gradient methods. This gap highlights the need for a unified framework that analyzes privacy guarantees and convergence rate for maximizing the privacy-utility trade-off in private shuffled gradient methods.

An intriguing extension of PABI involves leveraging public data samples to amplify

privacy. By performing gradient steps on s private samples followed by m public samples, the privacy amplification factor improves by approximately $\frac{1}{m}$, as sensitive information from private samples, where were used at the beginning of the optimization procedure, is less exposed in the final model. However, practical challenges arise when public and private data distributions differ. While using more public samples enhances privacy amplification, it may degrade convergence performance due to distribution mismatch. This trade-off underscores the need for strategies to effectively utilize public data in private reshuffled gradient methods.

4.1.1 Our Contributions

In this work, we aim to enhance the understanding of private shuffled gradient methods by focusing on the most basic variant: the Incremental Gradient (IG) method. Our contributions are threefold:

1. **Generalized IG Framework.** To study private IG with the potential usage of public data, we propose the Generalized IG Framework based on the last-iterate analysis, which enables optimization using a surrogate objective function and noise injection for privacy purpose. Here, the surrogate objective is defined by the public samples, and the true objective is defined by the private samples. To analyze the Generalized IG Framework, we propose a new metric of measuring the dissimilarity between the true and the surrogate objective, specifically tailored to the context of IG.
2. **Understanding DP-IG.** DP-IG is a special case in the proposed Generalized IG Framework, where only the true objective function is used across all epochs. We give the first convergence rate and show that DP-IG achieves an empirical excess risk of $\mathcal{O}(\frac{(\sqrt{d})^{4/3}}{n^{2/3}\epsilon^{4/3}})$ with a privacy analysis based on PABI. This is in contrast to the existing lower bound of $\mathcal{O}(\frac{\sqrt{d}}{n\epsilon})$. We give a detailed discussion comparing and explain the difference in the empirical excess risk of DP-IG, DP-SGD and DP-GD.
3. **Effective Usage of Public Data to Maximize Privacy-Utility Trade-offs.** Based on the Generalized IG Framework, we show that an interleaved optimization using private and public data samples leads to the best convergence rate compared to other baselines, given a reasonable dissimilarity between the

private and public data samples.

4. **Validation in Experiments.** Lastly, we empirically compare different options of using public and private samples in private IG, with different dissimilarity between them in various applications and datasets.

4.2 Preliminary

4.2.1 Related Work

Shuffled Gradient Methods. While the convergence rate of SGD in non-private settings is well established, understanding the convergence behavior of shuffled gradient methods, particularly Random Reshuffling (RR), has been a more recent development. Significant advances include characterizing the convergence rate of RR [78, 79] and establishing last-iterate convergence results for shuffled gradient methods applied to composite objective functions [69].

Privacy Amplification by Iteration (PABI). In practical applications, only the last-iterate model parameter is typically used during inference, while intermediate versions generated during optimization are discarded. However, current privacy analyses of DP-SGD often assume that all intermediate model parameters are released. This discrepancy has motivated a line of research investigating the privacy loss of the last-iterate model parameter under the assumption that intermediate parameters remain private [7, 36, 121]. This line of work shows privacy amplification by iteration (PABI).

Most existing works on PABI, however, emphasize privacy guarantees without exploring their implications for convergence. One exception is [36], which contextualizes privacy analysis in private optimization for stochastic convex optimization (SCO) problems though, rather than empirical risk minimization (ERM). Their convergence analysis relies on existing average-iterate bounds, which are inconsistent with PABI's focus on last-iterate privacy. To bridge this gap, the authors analyze impractical variants of DP-SGD, such as algorithms that randomly skip or terminate after a randomly chosen number of gradient steps. While they briefly mention the use of public data in private optimization, they do not address more realistic scenarios where public and private datasets follow different distributions.

In a related study, [7] demonstrates that DP-SGD applied to convex, smooth, and Lipschitz objectives with bounded domains incurs a finite privacy loss, rather an infinite privacy loss as privacy composition indicates. However, their analysis critically depends on i.i.d. sampling in DP-SGD and the assumption that all model parameters remain within a bounded domain at every gradient step. These conditions differ significantly from those of shuffled gradient methods, making their results inapplicable to this setting. [121] shows that in a more restricted setting where the objective function is strongly convex, even without a bounded domain, the privacy loss is finite.

Private Optimization. The privacy loss of DP-SGD based on privacy amplification by subsampling, is well understood [1], including a tight upper and lower bound for solving empirical risk minimization problems [12]. However, recent work has observed the gap between theory and practice: people use shuffled gradient methods, while computing the amount of noise based on the analysis of DP-SGD [22, 23]. However, this line of work focuses on privacy only, and there is no unified anlayze from both optimization and privacy.

Another line of research addresses the impractical i.i.d. subsampling assumption in DP-SGD by introducing correlated noise across iterations, e.g., the seminal work in [55] and its follow-up work on matrix factorization mechanisms [21], which optimizes the constants in the privacy analysis using correlated noise. While this direction improves the practicality of DP mechanisms, it does not directly address the unique convergence behavior of shuffled gradient methods, which are still widely used in practice [22] despite this line of work.

Other Related Work. Other works in optimization and privacy have explored various aspects, but significant gaps remain in understanding private shuffled gradient methods. For instance, the use of surrogate objectives in optimization, as discussed in [117], highlights potential applications in private optimization. However, this approach does not include a privacy analysis and does not employ shuffled gradient methods. Similarly, prior studies leveraging public data in private learning or optimization have focused on maximizing the privacy-utility trade-offs [16, 113]. Despite these advancements, none of these works address private shuffled gradient methods or consider the application of PABI in privacy analysis.

4.2.2 Background and Notation

Differential Privacy

Definition 4.2.1 (Differential Privacy (DP) [32]). *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with a domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy for $\epsilon, \delta \geq 0$ if for any two* **adjacent datasets** *$\mathcal{D}, \mathcal{D}'$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that*

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

$\delta = 0$ is often called pure differential privacy; while $\delta > 0$ is often called approximate differential privacy.

Definition 4.2.2 (Renyi Divergence). *For two probability distributions P and Q defined over $\mathcal{R}$, the Renyi divergence of order $\alpha > 1$ is*

$$PQ := \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \Big(\frac{P(x)}{Q(x)} \Big)^{\alpha}$$

Definition 4.2.3 ((α, ϵ)-Renyi Differential Privacy (RDP) [77]). *A randomized mechanism $f : \mathcal{D} \to \mathcal{R}$ is said to have ϵ-Renyi differential privacy of order α, or (α, ϵ)-RDP for short, if for any adjacent $D, D' \in \mathcal{D}$, it holds that*

$$f(D) f(D') \leq \epsilon$$

Proposition 4.2.4 (From RDP to DP (Proposition 3 of [77])). *If f is an (α, ϵ)-RDP mechanism, it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP for any $0 < \delta < 1$.*

Proposition 4.2.5 (RDP Composition (Proposition 1 of [77])). *Let $f : \mathcal{D} \to \mathcal{R}_1$ be (α, ϵ_1)-RDP and $g : \mathcal{R}_1 \times \mathcal{D} \to \mathcal{R}_2$ be (α, ϵ_2)-RDP, then the mechanism defined as (X, Y), where $X \sim f(D)$ and $Y \sim g(X, D)$, satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

Privacy Amplification by Iteration (PABI)

Definition 4.2.6 (Contraction (Definition)). *For a Banach space $(\mathcal{Z}, \|\cdot\|)$ A function $\psi : \mathcal{Z} \to \mathcal{Z}$ is said to be contractive if it is 1-Lipschitz, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{Z}$, $\|\psi(\mathbf{x}) - \psi(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$.*

Remark 4.2.7. *As shown in [36], taking one gradient step of a convex and L-smooth objective f, i.e., $\psi(\mathbf{x}) = \mathbf{x} - \eta\nabla_{\mathbf{x}} f(\mathbf{x})$, for which $\eta \leq 2/L$ and the projection operator $\Pi_{\mathcal{Q}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y}\in\mathcal{Q}} \|\mathbf{x} - \mathbf{y}\|$ onto a convex set $\mathcal{Q}$ are both contractive.*

Definition 4.2.8 (Contractive Noisy Iteration (Definition 19 of [36]))**.** *Given a random initial state $X_0 \in \mathcal{Z}$, a sequence of contractive functions $\psi_t : \mathcal{Z} \to \mathcal{Z}$, and a sequence of noise distribution $\{\rho_t\}$, the contractive noisy iteration (CNI) is defined by the following update rule:*

$$X_{t+1} = \psi_{t+1}(X_t) + Z_{t+1}$$

where Z_{t+1} is drawn independently from ρ_{t+1}. The random variable output by this process after T steps is denoted as $CNI(X_0, \{\psi_t\}, \{\rho_t\})$.

Theorem 4.2.9 (Privacy Amplification by Iteration (Theorem 22 of [36] with Gaussian Noise))**.** *Let X_T and X'_T denote the output of $CNI_T(X_0, \{\psi_t\}, \{\rho_t\})$ and $CNI_T(X_0, \{\psi'_t\}, \{\rho_t\})$. Let $s_t := \sup_{\mathbf{x}} \|\psi_t(\mathbf{x}) - \psi'_t(\mathbf{x})\|$, where $\rho_t \sim \mathcal{N}(0, \sigma^2\mathbb{I}_d)$ for all t. Let $a_1, \ldots, a_T$ be a sequence of reals and let $z_t := \sum_{i\leq t} s_i - \sum_{i\leq t} a_i$. If $z_t \geq 0$ for all t and $z_T = 0$, then*

$$X_T X'_T \leq \sum_{t=1}^{T} \frac{\alpha a_t^2}{2\sigma^2}$$

Notation and Problem

We consider the constrained empirical risk minimization problem with the true objective defined over a private dataset $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$, where $\mathbf{d}_i \in \mathbb{R}^d, \forall i \in [n]$, as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \Big\{ F(\mathbf{x}) + \psi(\mathbf{x}) \Big\}, \text{ where } F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \mathbf{d}_i) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

where $\mathcal{Q} \subseteq \mathbb{R}^d$ is a convex set. $\psi : \mathcal{Q} \to \mathbb{R}$ is a regularization function. We enforce $\psi = \mathcal{I}\{\mathbf{x} \in \mathcal{Q}\} + g(\mathbf{x})$ for some function $g : \mathcal{Q} \to \mathbb{R}$ to ensure $\mathbf{x} \in \mathcal{Q}$.

We also consider cases where a surrogate dataset $\Psi = \{\psi_1, \ldots, \psi_n\}$, where $\psi_i \in \mathbb{R}^d$,

is available, and define the surrogate objective one might use in optimization as

$$\min_{\mathbf{x}\in\mathbb{R}^d}\Big\{\widetilde{F}(\mathbf{x})+\psi(\mathbf{x})\Big\},\ \text{where } \widetilde{F}:=\frac{1}{n}\sum_{i=1}^{n}f(\mathbf{x}_i;\psi_i)=\frac{1}{n}\sum_{i=1}^{n}\widetilde{f}_i(\mathbf{x})$$

Furthermore, for analysis purpose, we defined the difference in the two objectives as

$$H(\mathbf{x})=F(\mathbf{x})-\widetilde{F}(\mathbf{x})=\frac{1}{n}\sum_{i=1}^{n}\Big(f_i(\mathbf{x})-\widetilde{f}_i(\mathbf{x})\Big)=\frac{1}{n}\sum_{i=1}^{n}h_i(\mathbf{x})$$

Let $\mathbf{x}^*:=\min_{\mathbf{x}\in\mathcal{Q}}F(\mathbf{x})$ denote the optimum of the true objective. Define the Bregman divergence induced by g as $B_g(\mathbf{x},\mathbf{y}):=g(\mathbf{x})-g(\mathbf{y})-\langle\nabla g(\mathbf{y}),\mathbf{x}-\mathbf{y}\rangle,\forall\mathbf{x},\mathbf{y}\in\mathbb{R}^d$. In the algorithm, we use the superscript $\cdot^{(s)}$ to denote a parameter used in the s-th epoch in optimization. For example, $F^{(s)}$ is the objective used in the s-th epoch. Moreover, let $L_i^{(s)}=\{L_i,\widetilde{L}_i\}$ denote the smoothness parameter of $F^{(s)}$, and define $L^{(s)}\in\{L,\widetilde{L}\}=\{\frac{1}{n}\sum_{i=1}^{n}L_i,\frac{1}{n}\sum_{i=1}^{n}\widetilde{L}_i\}$, $L^{(s)*}\in\{L^*,\widetilde{L}^*\}=\{\max\{L_i\}_{i=1}^n,\max\{\widetilde{L}_i\}_{i=1}^n\}$. $\|\cdot\|$ denotes the ℓ_2 norm. $\mathcal{I}\{\cdot\}$ is the indicator function, and $\mathbb{I}_d$ is the identity matrix in d dimension.

The utility metric we use is the classical convergence rate metric under a convex objective function F: $\mathbb{E}[F(\widetilde{\mathbf{x}})]-\mathbb{E}[F(\mathbf{x}^*)]$, where $\widetilde{\mathbf{x}}$ is the output of the optimization algorithm and $\mathbf{x}^*=\operatorname{argmin}_{\mathbf{x}\in\mathcal{Q}}F(\mathbf{x})$ is the optimum. We refer to this metric as the "empirical excess risk" if the learning rate η and the number of epochs K are both chosen to maximize the convergence rate, which measures the irreducible gap between the optimal objective value and the minimal objective value by the algorithm.

4.3 Generalized IG Framework

4.3.1 Basic Assumptions

We start with the following common assumptions in optimization and privacy analysis. We note that in order to use PABI to improve privacy-convergence rate trade-off, each component of the function used in optimization, i.e., the true objective or the surrogate, needs to satisfy convexity, Lipschitzness and smoothness simultaneously. It

is possible to remove the Lipschitzness condition by using gradient clipping techniques (see, e.g., [35]). To show convergence, each component of the function used in optimization only needs to be convex and smooth. Here, the smoothness condition can be replaced by Lipschitzness, but this leads to a slower convergence rate of $\mathcal{O}(\frac{1}{\sqrt{K}})$ for K epochs. See [69] for more details.

Assumption 4.3.1 (Convexity)**.** *f_i is convex, $\forall i \in [n]$, and $\widetilde{f}_j$ is convex, $\forall j \in [m]$.*

Assumption 4.3.2 (Smoothnessness)**.** *f_i is L_i-smooth, $\forall i \in [n]$, and $\widetilde{f}_j$ is $\widetilde{L}_j$-smooth, $\forall j \in [m]$. That is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$, $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L_i\|\mathbf{x} - \mathbf{y}\|$ and $\|\nabla \widetilde{f}_j(\mathbf{x}) - \nabla \widetilde{f}_j(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$.*

Assumption 4.3.3 (Lipschitzeness)**.** *f_i is G_i-Lipschitz continuous, $\forall i \in [n]$, and $\widetilde{f}_i$ is $\widetilde{G}_j$-Lipschitz continuous, $\forall j \in [m]$. That is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$, $\|f_i(\mathbf{x}) - f_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$ and $\|\widetilde{f}_j(\mathbf{x}) - \widetilde{f}_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$.*

Assumption 4.3.4 (Regularization)**.** *The regularization function $\psi : \mathcal{Q} \to \mathbb{R}^d$ is μ_ψ-strongly convex, for $\mu_\psi \geq 0$. For example, if $\psi(\mathbf{x}) = \mathcal{I}\{\mathbf{x} \in \mathcal{Q}\}$ and $\mathcal{Q} \subseteq \mathbb{R}^d$ is a convex set, then $\mu_\psi = 0$.*

Assumption 4.3.5 (Uncertainty in Optimization)**.** *The uncertainty induced by shuffling in optimization is $\sigma_{any}^2 = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}^*)$.*

4.3.2 Measuring Dissimilarity between the True Objective and the Surrogate

The basic assumptions imply that each component of the difference function h_i is $(L_i + \widetilde{L}_i)$-smooth and $(G_i + \widetilde{G}_i)$-Lipschitz. However, these parameters $(L_i + \widetilde{L}_i)$ and $(G_i + \widetilde{G}_i)$ are too pessimistic. For example, when $F = \widetilde{F}$ (no dissimilarity), $H = 0$ and H is 0-smooth, 0-Lipschitz continuous. Intuitively, smaller smoothness and Lipschitz parameters indicate better convergence towards F, since there is less dissimilarity between the true objective F and the surrogate $\widetilde{F}$. Hence, to capture such dissimilarity, we explicitly introduce the smoothness and Lipschitzness parameters of H.

Assumption 4.3.6 (Smoothness of H)**.** *H is L_H-smooth, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$, $\|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\| \leq L_H\|\mathbf{x} - \mathbf{y}\|$.*

The following notion of dissimilarity (with a connection to Lipschitzness of H), is inspired by dissimilarity measures in federated learning with heterogeneous clients (see, e.g., [112]), and the prior work on using a surrogate objective [117] in SGD.

However, in IG, we propose a more fine-grained notion of dissimilarity that include partial dissimilarity of samples in the order and avoids defining the dissimilarity by comparing the gradient of the objective function evaluated at individual samples $(\mathbf{d}_i, \psi_i), \forall i \in [n]$. As we see in the following remarks, this is necessary for tighter dissimilarity measures in certain cases.

Assumption 4.3.7 (Dissimilarity)**.** *For all* $\mathbf{x} \in \mathcal{Q}$ *and* $i \in [n]$, $\|\sum_{j=1}^{i} \nabla h_j(\mathbf{x})\| \leq C_i, \forall i \in [n]$.

Remark 4.3.8. *C_i is guaranteed to exist, $\forall i \in [n]$, since $\|\sum_{j=1}^{i} \nabla h_j(\mathbf{x})\| \leq \sum_{j=1}^{i}(G_i + \widetilde{G}_i)$, as implied by Assumption 4.3.3. However, as mentioned above, $G_i + \widetilde{G}_i$ can be quite loose. For example, consider the case where $\psi_i = \mathbf{d}_{i+1}, \forall i \in [n-1]$ and $\psi_n = \mathbf{d}_1$, i.e., Ψ is a permutation of $\mathcal{D}$. Furthermore, for simplicity, let $G_i := G, \widetilde{G}_i := \widetilde{G}, \forall i \in [n]$. Then, Assumption 4.3.7 implies a tighter $C_i \leq G + \widetilde{G}$, $\forall i \in [n]$, instead of the naïve bound $C_i \leq i(G + \widetilde{G})$.*

Remark 4.3.9. *One might attempt to define the dissimilarity measure as $\frac{1}{n}\sum_{j=1}^{n} \|\nabla h_j(\mathbf{x})\| \leq C$, which is a stronger version of Assumption 4.3.7, due to Jensen's inequality. But this definition can lead to a very loose bound on dissimilarity. For example, consider the permutation case again, this notion of dissimilarity measure suggests $\|\frac{1}{n}\sum_{j=1}^{n} \nabla h_j(\mathbf{x})\| \leq \frac{1}{n}\sum_{j=1}^{n} \|\nabla h_j(\mathbf{x})\| \leq G$, but indeed one should have $\frac{1}{n}\|\sum_{j=1}^{n} \nabla h_j(\mathbf{x})\| = 0$. Assumption 4.3.7, however, is able to accurately reflect the 0 dissimilarity in this case.*

Remark 4.3.10. *Moreover, Assumption 4.3.7 reflects the fact that the order of the samples used in IG leads to different optimization trajectories, but as long as the surrogate dataset Ψ and the true dataset $\mathcal{D}$ contain similar data samples, i.e., small $\|\sum_{i=1}^{n} \nabla h_i(\mathbf{x})\|$, optimization using $\mathcal{D}$ and using Ψ shall converge to similar objective values $F(\mathbf{x})$'s. For example, consider the permutation case again, though optimizing using $\mathcal{D}$ and using $\mathcal{P}$ leads to different trajectories (i.e., intermediate $\mathbf{x}$'s), but intuitively, both optimization procedures should converge to the same point, because the data samples are essentially the same. As we see in the next section, Assumption 4.3.7 captures both the difference in trajectories and the fact that both optimization procedures essentially converge to the same $F(\mathbf{x})$.*

4.3.3 Algorithm and Convergence Rate

We are now ready to present the generalized framework of IG in Algorithm 2, which enables the usage of surrogate objectives and the injection of noise for privacy purpose during the optimization procedure. Note one can choose to use different surrogate objectives $F^{(s)} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \psi_i^{(s)}) = \frac{1}{n}\sum_{i=1}^{n} \widetilde{f}_i^{(s)}(\mathbf{x})$ defined over different surrogate datasets $\Psi^{(s)} = \{\psi_1^{(s)}, \ldots, \psi_n^{(s)}\}$ in different epochs $s \in [K]$. This means the difference function $H^{(s)} = F - \widetilde{F}^{(s)}$ can be different at different epochs s. We denote the smoothness parameter of $H^{(s)}$ as $L_H^{(s)}$.

Algorithm 2 Generalized IG

Input: Initial point $\mathbf{x}_1^{(1)}$, learning rate η, convex domain set $\mathcal{Q} \subseteq \mathbb{R}^d$. Sequence of objective functions $\{F^{(s)}\}_{s=1}^K$. Sequence of noise variance $\{(\sigma_z^{(s)})^2\}_{s=1}^K$. Regularization function $\psi = \mathcal{I}\{\mathbf{x} \in \mathcal{Q}\} + g$ for $g : \mathcal{Q} \to \mathbb{R}$.
for $s = 1, 2, \ldots, K$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    Sample noise $\rho_i^{(s)} \sim \mathcal{N}(0, (\sigma^{(s)})^2 \mathbb{I}_d)$
    $\mathbf{x}_{i+1}^{(s)} \leftarrow \mathbf{x}_i^{(s)} - \eta\left(\nabla f_i^{(s)}(\mathbf{x}_i^{(s)}) + \rho_i^{(s)}\right)$
  end for
  $\mathbf{x}_1^{(s+1)} \leftarrow \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} n\psi(\mathbf{x}) + \frac{\|\mathbf{x}-\mathbf{x}_{n+1}^{(s)}\|^2}{2\eta}$
end for
return $\mathbf{x}_1^{(K+1)}$

Theorem 4.3.11 (Convergence Rate for K Epochs)**.** *Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $k \in [K]$ and a hyperparameter $\beta > 0$, if $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$, and constant learning rate $\eta \leq \frac{1}{2n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}(1+\log K)}}$, Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \qquad (4.1)\\
&\leq \frac{1}{\eta n K}\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K)\max\{L, \widetilde{L}\}\\
&\quad + \underbrace{\frac{1}{2\beta}\sum_{s=1}^{K}\frac{1}{K+1-s}(C_n^{(s)})^2}_{\textit{Non-vanishing Dissimilarity}} + \underbrace{5\eta^2 \cdot \sum_{s=1}^{K}\frac{1}{K+1-s}\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2}_{\textit{Vanishing Dissimilarity}}
\end{aligned}$$

$$+ \underbrace{5\eta^2 nd \sum_{s=1}^{K} \frac{1}{K+1-s} L^{(s)} (\sigma^{(s)})^2}_{\textit{Injected Noise}}$$

where the expectation is taken w.r.t. the injected noise.

See Appendix C.1 for the proof of Theorem 4.3.11.

Remark 4.3.12. *The above result recovers the convergence rate of IG in non-private settings in [69] under the same set of assumptions with no dissimilarity, i.e.,* $\widetilde{L} = L$, $L^{(s)} = L$, $C_i^{(s)} = 0$, *and* $C^{(s)} = 0$, $\forall s \in [K]$ *and no noise injection, i.e.,* $(\sigma^{(s)})^2 = 0, \forall s \in [K]$.

The main difference between the above convergence rate bound in Eq. 4.1 and the vanilla IG bound in non-private settings is the three additional terms: 1) the non-vanishing dissimilarity term, 2) the vanishing dissimilarity term, and 3) the term due to injected noise. We illustrate the difference in the non-vanishing and the vanishing dissimilarity terms using the following example.

Remark 4.3.13. *Consider again when the surrogate* Ψ *is a permutation of the true dataset* $\mathcal{D}$, *where* $\psi_i = \mathbf{d}_{i+1}, \forall i \in [n-1]$ *and* $\psi_n = \mathbf{d}_1$, *and we use the surrogate* $F^{(s)}(\mathbf{x}) = \widetilde{F}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \psi_i)$ *in optimization for all epochs* $s \in [K]$. *For simplicity, suppose the Lipschitz parameter in Assumption 4.3.3 is* $G_i := G, \forall i \in [n]$, *which also implies* $\widetilde{G}_i = G$, *and the smoothness parameter in Assumption 4.3.2 is* $L_i := L, \forall i \in [n]$, *which implies* $\widetilde{L}_i = L$. *Now, the dissimilarity measure is* $C_i \leq 2G, \forall i \in [n-1]$ *and* $C_n = 0$. *In the above convergence rate bound, the term* $5\eta^2 \cdot \sum_{s=1}^{K} \frac{1}{K+1-s} \frac{1}{n} \sum_{i=1}^{n-1} L_{i+1} (C_i^{(s)})^2 = 5\eta^2 \cdot \sum_{s=1}^{K} \frac{1}{K+1-s} \frac{1}{n} \sum_{i=1}^{n-1} L 4G^2 \leq 20\eta^2 (1 + \log K) \frac{n-1}{n} LG^2$ *measures the deviation of the trajectory by optimizing using* Ψ *than using the true dataset* $\mathcal{D}$. *However, this term scales with* η^2, *and* η *is often set to be* $\propto \frac{1}{K^{2/3}}$ *in IG, meaning that such deviation in trajectory vanishes as the number of epoch increases. Furthermore, the non-vanishing term due to dissimilarity is* $\frac{1}{2\beta} \sum_{s=1}^{K} \frac{1}{K+1-s} (C_n^{(s)})^2 = 0$, *due to* $C_n^{(s)} = C_n = 0$. *Hence, optimizing using the surrogate dataset* Ψ *and optimizing using the true dataset* $\mathcal{D}$ *converges to the same* $F(\mathbf{x})$ *eventually, when* Ψ *is a permutation of* $\mathcal{D}$, *even though they have different trajectories. However, with a different surrogate dataset* Ψ *that might not be a permutatino of* $\mathcal{D}$, *it is not guaranteed that* $C_n = 0$, *and this leads to a non-vanishing term due to objective dissimilarity. Note this non-vanishing term also exists in*

cases such as SGD with a surrogate function [117], or federated learning with client heterogeneity [112].

4.4 Private IG Using Private Data Only

Algorithm 2 enables us to analyze a differentially private version of IG, which we call DP-IG, by using the true objective with private samples only. To the best of our knowledge, this is the first analysis of private shuffled gradient methods.

In DP-IG we use the true objective across all epochs, i.e., $F^{(s)} = F, \forall s \in [K]$, and this implies there is no dissimilarity, i.e., $C_i^{(s)} = 0$, $\forall s \in [K], i \in [n]$. The injected Gaussian noise has variance $(\sigma^{(s)})^2 = \sigma^2, \forall s \in [K]$ for some σ^2 we set later to guarantee a maximum privacy loss over all epochs. This leads to the following convergence rate of DP-IG:

Corollary 4.4.1 (Convergence Rate of DP-IG). *Set $F^{(s)} = F$ and $(\sigma^{(s)})^2 = \sigma^2$, $\forall s \in [K]$ in Algorithm 2. Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $k \in [K]$ and a hyperparameter $\beta > 0$, if $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$, and $\eta \leq \frac{1}{2nL^*\sqrt{10(1+\log K)}}$, Algorithm 2 guarantees*

$$\begin{aligned} &\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \\ &\leq \frac{\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2}{\eta n K} + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K) L + 5\eta^2 n d L (1+\log K)\sigma^2 \end{aligned} \tag{4.2}$$

Privacy. Within each epoch, since the learning rate is $\eta \leq \frac{1}{L^*}$, taking a gradient step $\mathbf{x}_{i+1}^{(s)} \leftarrow \mathbf{x}_i^{(s)} - \eta \nabla f_i^{(s)}(\mathbf{x}_i^{(s)})$ is "contractive" (see Remark 4.2.7), and since the intermediate variables $\mathbf{x}_i^{(s)}$ does not need to be released, we can apply PABI (see Theorem 4.2.9) to reason about the privacy loss per epoch. The regularization step at the end of each epoch in line 7 of Algorithm 2, however, is not "contractive". This prevents us from applying PABI across epochs. We use composition theorem of RDP (see Property 4.2.5) to compute the privacy loss across epochs.

Lemma 4.4.2 (Privacy Guarantee of DP-IG). *The output $\mathbf{x}_1^{(K+1)}$ of DP-IG is $(\alpha, \frac{2\alpha(G^*)^2 K}{\sigma^2})$-RDP for $\alpha > 1$, that is, $(\frac{2\alpha(G^*)^2 K}{\sigma^2} + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP, for $\alpha > 1, \delta > 0$.*

We optimize α to minimize the total privacy loss such that $\alpha = \frac{\sigma\sqrt{\log 1/\delta}}{G^*\sqrt{2K}}$. The

output of DP-IG is then $(\mathcal{O}(\frac{G^*\sqrt{K\log 1/\delta}}{\sigma}), \delta)$-DP. We fix the overall privacy loss to be ϵ, such that $\frac{G^*\sqrt{K\log 1/\delta}}{\sigma} = \epsilon$. This implies the amount of noise needed in DP-IG is $\sigma = \mathcal{O}\Big(\frac{G^*\sqrt{K\log 1/\delta}}{\epsilon}\Big)$.

Choosing the Learning Rate. To minimize the convergence rate in Eq. 4.2, let $\widetilde{\sigma}^2 = n^2\sigma_{any}^2 + nd\sigma^2$, where $\sigma^2 = \frac{(G^*)^2 K\log 1/\delta}{\epsilon^2}$, ignoring all constants, and set $\frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^2}{\eta nK} = \eta^2 L(1+\log K)\widetilde{\sigma}^2$, i.e., $\eta = \frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^{2/3}}{(nKL(1+\log K)\widetilde{\sigma}^2)^{1/3}}$. This results in a convergence rate of

$$\mathcal{O}\Big(\frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^{4/3}L^{1/3}(1+\log K)^{1/3}\sigma_{any}^{2/3}}{K^{2/3}} + \frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^{4/3}L^{1/3}(1+\log K)^{1/3}G^{2/3}\log^{1/3}(1/\delta)\cdot d^{1/3}}{K^{1/3}n^{1/3}\epsilon^{1/3}}\Big)$$

Empirical Excess Risk. We now choose the number of epochs, K, get the empirical empirical excess risk, specifically in terms of the dependency on n, d, ϵ. To minimize the empirical excess risk, set $\frac{1}{K^{2/3}} = \frac{d^{1/3}}{K^{1/3}n^{1/3}\epsilon^{1/3}}$, i.e., $K = \frac{n\epsilon^2}{d}$. And the resulting empirical excess risk of DP-IG is $\mathcal{O}\Big(\frac{1}{n^{2/3}}(\frac{\sqrt{d}}{\epsilon})^{\frac{4}{3}}\Big)$.

DP-IG vs. Lower Bound. The lower bound of empirical excess risk for optimization problems with smooth, convex objectives in a bounded domain is $\Omega\Big(\frac{\sqrt{d}}{n\epsilon}\Big)$ [12]. DP-IG's empirical excess risk bound is larger than the lower bound in many regimes of interest when $n^{1/3}(\frac{\sqrt{d}}{\epsilon})^{1/3} > 1$.

DP-IG vs. DP-GD. The convergence rate of DP-GD is $\mathcal{O}\Big(\frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^2}{\eta T} + \eta\widetilde{\sigma}\Big)$ for T steps of gradient computation [3], where σ is the std. of the injected Gaussian noise. By the privacy composition theorem, $\sigma = \frac{G^*\sqrt{T\log 1/\delta}}{n\epsilon}$. We compare the empirical excess risk of DP-IG and DP-SGD by fixing the total number of gradient steps to be $T = Kn$. To enable a more straightforward comparison and a clean decomposition of the convergence rate into terms due to optimization and terms due to the injected noise, we intentionally choose $\eta = \frac{1}{nL}$ for both algorithms, independent of T or K. Doing so does not change the empirical excess risk of DP-SGD yet slightly increases the empirical excess risk of DP-IG but enables a more straightforward comparison. Setting η as such also increases T or K required to achieve the minimum empirical excess risk but this is not the focus of comparison.

Table 4.1 presents a comparison of DP-IG and DP-GD, whose empirical excess risk matches the lower bound. The second column shows that DP-IG has a larger empirical excess risk by a factor of $\frac{1}{\sqrt{n}}$ compared to DP-GD. The reason of such a

Algorithm	η	Convergence Rate	Empirical Excess Risk
DP-GD	$\frac{1}{nL}$	$\mathcal{O}\Big(\underbrace{\frac{\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 nL}{T}}_{\text{Dep. on Initialization}} + \underbrace{\frac{d}{nL} \cdot \frac{(G^*)^2 T \log 1/\delta}{n^2\epsilon^2}}_{\text{Injected Noise}} \Big)$	$\widetilde{\mathcal{O}}\Big(\frac{\sqrt{d}}{n\epsilon}\Big)$ ‡
DP-IG		$\underbrace{\widetilde{O}\Big(\frac{\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 L}{K}\Big)}_{\text{Dep. on Initialization}} + \underbrace{\frac{\sigma_{any}^2}{L}}_{\text{Opt. Uncertainty}} + \underbrace{\frac{d}{nL} \cdot \frac{(G^*)^2 K \log 1/\delta}{\epsilon^2}\Big)}_{\text{Injected Noise}}$ †	$\widetilde{\mathcal{O}}\Big(\frac{\sqrt{d}}{\sqrt{n}\epsilon}\Big)$ ‡

† $\widetilde{\mathcal{O}}$ suppresses logarithmic terms in K.
‡ $\widetilde{\mathcal{O}}$ suppresses logarithmic terms in $n, d, \epsilon, 1/\delta$.

Table 4.1: Comparing the convergence rate and empirical excess risk of DP-IG and DP-GD

difference is explained in the rest of the table, where in DP-GD the sensitivity of the gradient is smaller by a factor of $\frac{1}{n}$ and we composes across T gradient steps; while in DP-IG the gradient sensitivity is G^*, and by using PABI, we only need to compose across $K = T/n$ epochs.

Algorithm	Gradient Sensitivity	# Composition	Privacy Loss
DP-GD	$\frac{G^*}{n}$	T	$\propto \frac{G}{n}\sqrt{T}$
DP-IG	G^*	$K = T/n$	$\propto G\sqrt{\frac{T}{n}}$

Table 4.2: Explaining the root cause of a $\frac{1}{\sqrt{n}}$ difference in the empirical excess risk of DP-IG and DP-GD

DP-IG vs. DP-SGD. Although the gradient sensitivity in DP-SGD is G^* instead of $\frac{G^*}{n}$ as in DP-GD, the i.i.d. sampling procedure in DP-SGD enables privacy amplification by subsampling, which results in a reduced privacy loss by a factor of $\frac{1}{n}$ per gradient, achieving a similar affect in terms of the per step privacy loss as in DP-GD. This enables DP-SGD also to achieve an empirical excess risk of $\mathcal{O}\Big(\frac{\sqrt{d}}{n\epsilon}\Big)$, matching the lower bound (up to logarithmic factors).

It is perhaps not surprising that DP-IG is not able to have an empirical excess risk that matches the lower bound. Intuitively, the lack of inherent randomness/optimization noise due to i.i.d. sampling in DP-IG makes it have a higher excess risk than DP-SGD. In the non-private settings, such lack of inherent randomness makes IG converges faster ($\mathcal{O}\Big(\frac{1}{K^{2/3}}\Big)$) compared to SGD ($\mathcal{O}\Big(\frac{1}{\sqrt{T}}\Big)$). But it seems in private settings, such randomness due to i.i.d. sampling helps with reducing the privacy

loss and results in less noise injection required for DP-SGD, leading to a factor of $\frac{1}{n}$ less empirical excess risk compared to DP-IG. It is not known whether an empirical excess risk of $\mathcal{O}\left(\frac{1}{n^{2/3}}(\frac{\sqrt{d}}{\epsilon})^{4/3}\right)$ is the best achivable one by DP-IG. We leave this as an interesting open question for future exploration.

4.5 Public Data Assisted DP-IG

Consider applications where a public dataset $\Psi = \{\psi_1, \ldots, \psi_n\}$ is available. If the public samples are from the same distribution as the private samples, one would optimize simply using the public samples. Hence, we consider the more practical case where the public dataset Ψ and the private dataset can be from different distributions. We discuss three different optimization algorithms making use of such public data, in the hope of adding less noise due to privacy constraints to maximize the utility-privacy trade-offs.

Define the surrogate objective as $\widetilde{F} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \psi_i) = \frac{1}{n}\sum_{i=1}^{n} \widetilde{f}_i(\mathbf{x})$. Let the dissimilarity measure be $\widetilde{C}^2 = \frac{1}{n}\sum_{i=1}^{n-1} \widetilde{L}_{i+1}(C_i)^2$.

4.5.1 Pub-Priv-IG

$K - S$ epochs of optimization on the public data using the surrogate objective $\widetilde{F}$, followed by S epochs of optimization on the private data using the true objective F. That is,

$$F^{(s)} = \begin{cases} \widetilde{F} & \text{if } s \leq S \\ F & \text{if } s > S \end{cases}, \quad (\sigma^{(s)})^2 = \begin{cases} 0 & \text{if } s \leq S \\ \sigma^2 & \text{if } s > S \end{cases}, \quad L^{(s)} = \begin{cases} \widetilde{L} & \text{if } s \leq S \\ L & \text{if } s > S \end{cases} \tag{4.3}$$

Furthermore, the dissimilarity measure is

$$C_n^{(s)} = \begin{cases} C & \text{if } s \leq S \\ 0 & \text{if } s > S \end{cases}, \quad \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 = \begin{cases} \widetilde{C}^2 & \text{if } s \leq S \\ 0 & \text{if } s > S \end{cases} \tag{4.4}$$

Corollary 4.5.1 (Convergence Rate of Pub-Priv-IG)**.** *Set $F^{(s)}$ and $(\sigma^{(s)})^2$ as in Eq. 4.3. Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $k \in [K]$ and a*

hyperparameter $\beta > 0$, *if* $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$, *and* $\eta \leq \frac{1}{2nL^*\sqrt{10(1+\log K)}}$, *Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \\
&\leq \frac{1}{\eta n K}\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K)\max\{L, \widetilde{L}\} \\
&\quad + \frac{1}{2\beta}(\log K - \log S)C_n^2 + 5\eta^2 \cdot (\log K - \log S)\widetilde{C}^2 + 5\eta^2 nd(1+\log S)L\sigma^2
\end{aligned}$$

Privacy. Just as in the DP-IG case, we can apply PABI (Theorem 4.2.9) within each epoch that involves optimization on the private dataset, and use the composition theorem of RDP (Theorem 4.2.5) to compute the total privacy loss across S epochs.

Lemma 4.5.2. *The output* $\mathbf{x}_1^{(K+1)}$ *of Pub-Priv-IG is* $(\alpha, \frac{2\alpha(G^*)^2 S}{\sigma^2})$*-RDP for* $\alpha > 1$*, that is,* $(\frac{2\alpha(G^*)^2 S}{\sigma^2} + \frac{\log 1/\delta}{\alpha - 1}, \delta)$*-DP, for* $\alpha > 1, \delta > 0$.

After choosing α to minimize the above total privacy loss across K epochs, one sets $\alpha = \frac{\sigma\sqrt{\log 1/\delta}}{G^*\sqrt{2S}}$, such that the output of Pub-Priv-IG is $(\mathcal{O}\Big(\frac{G^*\sqrt{K\log 1/\delta}}{\sigma}, \delta\Big), \delta)$-DP. Let $\frac{G^*\sqrt{K\log 1/\delta}}{\sigma} = \epsilon$, and so $\sigma = \mathcal{O}\Big(\frac{G^*\sqrt{S\log 1/\delta}}{\epsilon}\Big)$.

4.5.2 Priv-Pub-IG

$S \in [K]$ epochs of optimization on the private using the true objective F, followed by $K - S$ epochs of optimization on the public dataset using the surrogate objective $\widetilde{F}$. That is,

$$F^{(s)} = \begin{cases} F & \text{if } s \leq S \\ \widetilde{F} & \text{if } s > S \end{cases}, \quad (\sigma^{(s)})^2 = \begin{cases} \sigma^2 & \text{if } s \leq S \\ 0 & \textbf{if } s > S \end{cases}, \quad L^{(s)} = \begin{cases} L & \text{if } s \leq S \\ \widetilde{L} & \text{if } s > S \end{cases} \tag{4.5}$$

Furthermore, the dissimilarity measure is

$$C_n^{(s)} = \begin{cases} 0 & \text{if } s \leq S \\ C & \text{if } s > S \end{cases}, \quad \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 = \begin{cases} 0 & \text{if } s \leq S \\ \widetilde{C}^2 & \text{if } s > S \end{cases} \tag{4.6}$$

Corollary 4.5.3 (Convergence Rate of Priv-Pub-IG)**.** *Set* $F^{(s)}$ *and* $(\sigma^{(s)})^2$ *as in Eq. 4.5. Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any* $k \in [K]$ *and a*

hyperparameter $\beta > 0$, *if* $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$, *and* $\eta \leq \frac{1}{2nL^*\sqrt{10(1+\log K)}}$, *Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \\
&\leq \frac{1}{\eta n K}\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K)\max\{L, \widetilde{L}\} \\
&\quad + \frac{1}{2\beta}(1+\log(K-S))C_n^2 + 5\eta^2 \cdot (1+\log(K-S))\widetilde{C}^2 + 5\eta^2 nd(\log K - \log(K-S))L\sigma^2
\end{aligned} \tag{4.7}$$

Privacy. The computation of the privacy loss is exactly the same as in Pub-Priv-IG. That is, we also set $\sigma = \mathcal{O}\left(\frac{G^*\sqrt{S\log 1/\delta}}{\epsilon}\right)$ in Priv-Pub-IG here.

4.5.3 Interleaved-IG

Consider using $m \in [n]$ samples of Ψ. For each epoch, optimize on the private dataset using tshe true objective f_i for $n-m$ steps, followed by $n-m$ steps on the public dataset using the surrogate objective $\widetilde{f}_j$. That is, let the surrogate objective be $\widetilde{F}(\mathbf{x}) = \frac{1}{n}\left(\sum_{i=1}^{n-m} f(\mathbf{x}; \mathbf{d}_i) + \sum_{j=1}^{m} f(\mathbf{x}; \psi_j)\right)$.

$$F^{(s)} = F, \quad (\sigma^{(s)})^2 = \sigma^2, \quad L^{(s)} := \widehat{L} = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{i=1}^{n-m} L_i + \sum_{j=1}^{m}\widetilde{L}_i\right) \tag{4.8}$$

The dissimilarity measure is

$$C_n^{(s)} = C_n, \forall s \in [K], \quad \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 = \frac{1}{n}\sum_{i=n-m+1}^{n-1} L_{i+1}C_i^2 := \widetilde{C}, \forall s \in [K]$$

Corollary 4.5.4 (Convergence Rate of Interleaved-IG)**.** *Set* $F^{(s)}$ *and* $(\sigma^{(s)})^2$ *as in Eq. 4.8. Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any* $k \in [K]$ *and a hyperparameter* $\beta > 0$, *if* $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$, *and* $\eta \leq \frac{1}{2nL^*\sqrt{10(1+\log K)}}$, *Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \\
&\leq \frac{1}{\eta n K}\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2 + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K)\max\{L, \widetilde{L}\}
\end{aligned} \tag{4.9}$$

$$+\frac{1}{2\beta}(1+\log K)C_n^2+5\eta^2(1+\log K)\widetilde{C}^2+5\eta^2 ndL(1+\log K)\sigma^2$$

Privacy. In this setting, we can apply PABI (Theorem 4.2.9) to further gain privacy amplification by a factor of $\frac{2}{n}$ by making use of the public samples in each epoch. After applying composition across K epochs, we get the following total privacy loss

Lemma 4.5.5. *The output* $\mathbf{x}_1^{(K+1)}$ *of Interleaved-IG is* $(\alpha, \frac{2\alpha G^2}{(n+1-m)\sigma^2})$*-RDP, for* $\alpha > 1$*, that is,* $(\frac{2\alpha G^2}{(n+1-m)\sigma^2}+\frac{\log 1/\delta}{\alpha-1}, \delta)$*-DP, for* $\alpha > 1, \delta > 0$*.*

Similar to previous cases, after choosing $\alpha = \frac{\sigma\sqrt{(n+1-m)\log 1/\delta}}{G^*\sqrt{K}}$ that minimizes the total privacy loss and set the total privacy loss to be ϵ, one gets $\sigma = \mathcal{O}\Big(\frac{G^*\sqrt{K\log 1/\delta}}{\sqrt{n+1-m}\cdot\epsilon}\Big)$ in Interleaved-IG.

4.5.4 Comparison in a Special Case

We compare the empirical excess risk of the above three algorithms making use of public data samples in the special case, where each algorithm performs half of the total number of gradient computation using public samples, and the remaining half using private samples.

That is, in both Pub-Priv-IG and Priv-Pub-IG, $S = \frac{K}{2}$ and in Interleaved-IG, $m = \frac{n}{2}$. Table 4.3 presents the learning rate η and the number of epochs K that maximize the convergence rate, and the resulting empirical excess risk from each algorithm after choosing η and K.

Observe that asymptotically, the empirical excess risk of Pub-Priv-IG and Priv-Pub-IG are the same, though the constants in the convergence rate bound might differ. Notice the additional privacy amplification factor of $\frac{1}{n}$ by PABI due to optimization on $\frac{n}{2}$ public samples in Interleaved-IG results in a reduced empirical excess risk by a factor of $\frac{1}{n^{2/3}}$ compared to Pub-Priv-IG and Priv-Pub-IG, except the non-vanishing error of $\widetilde{O}(\frac{C_n^2}{\beta})$ due to dissimilarity by using a surrogate objective.

Algorithm	η	K	Empirical Excess Risk
Pub-Priv-IG Priv-Pub-IG	$\mathcal{O}\Big(\frac{\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^{2/3}}{\left[nK(1+\log K)(L\widetilde{\sigma}^2+\widetilde{C}^2)\right]^{1/3}}\Big)$ †	$\frac{n\epsilon^2}{d}$	$\widetilde{\mathcal{O}}\Big(\widetilde{C}^{2/3}(\frac{\sqrt{d}}{n\epsilon})^{4/3}+\frac{C_n^2}{\beta}+\frac{1}{n^{2/3}}(\frac{\sqrt{d}}{\epsilon})^{4/3}\Big)$ ‡
Interleaved-IG		$\frac{n^2\epsilon^2}{d}$	$\widetilde{\mathcal{O}}\Big(\widetilde{C}^{2/3}\frac{1}{n^2}(\frac{\sqrt{d}}{\epsilon})^{4/3}+\frac{C_n^2}{\beta}+(\frac{\sqrt{d}}{n\epsilon})^{4/3}\Big)$ ‡

† $\widetilde{\sigma}^2 = n^2\sigma_{any}^2 + nd\sigma^2$. σ^2 and $\widetilde{C}^2$ are both algorithm dependent.

‡ $\widetilde{\mathcal{O}}$ suppresses logarithmic factors in $n, d, \epsilon, 1/\delta$.

Table 4.3: Comparison of Empirical Excess Risk in terms of dependency on n, d, ϵ of three private optimization algorithms in the special case where half of gradient computation uses public samples.

4.6 Experiments: Public Data Assisted Private IG

In this section, we empirically demonstrate the difference in convergence between three different algorithm making use of public data: Pub-Priv-IG, Priv-Pub-IG and Interleaved-IG, where half of the gradient steps are computed on public samples. We also compare against two baselines: Private-IG, which uses private samples throughout all epochs; and Public-IG, which uses public samples without noise addition throughout all epochs.

Task: Mean Estimation. We focus on the task of mean estimation. Given the private training dataset $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$, where $\mathbf{d}_i \sim \mathcal{N}(0, \mathbb{I}_d)$, the true objective function is defined as

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x} - \mathbf{d}_i\|_2^2$$

Furthermore, we are given a public dataset consisting of n public samples $\Psi = \{\psi_1, \ldots, \psi_n\}$, where $\psi_j \sim \mathcal{N}(0.5, \mathbb{I}_d)$. The following surrogate objective function is used by Priv-Pub-IG and Pub-Priv-IG for $\frac{K}{2}$ epochs:

$$\widetilde{F}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x} - \psi_i\|^2$$

The following surrogate objective in Interleaved-IG for all epochs:

$$\widetilde{F}(\mathbf{x}) = \frac{1}{n}\Big(\sum_{i=1}^{n/2} \|\mathbf{x} - \mathbf{d}_i\|^2 + \sum_{j=1}^{n/2} \|\mathbf{x} - \psi_j\|^2\Big)$$

Hyperparameters. We do a grid search of the learning rate $\eta \in \{10^{-2}, 10^{-3}, \ldots, 10^{-7}\}$. The privacy parameters are $\delta = 10^{-6}$ and $\epsilon \in \{5, 10\}$.

Results. We begin by evaluating optimization performance using only public samples, without incorporating private samples or adding noise. The results of the Public-IG algorithm, along with comparisons to other methods, are shown in Figure 4.1. Next, we compare algorithms that utilize private samples. We repeat each experiment for 10 random runs. The results are presented in Figure 4.2, where the solid line indicates the mean of objective function values and the shaded area represents on std.

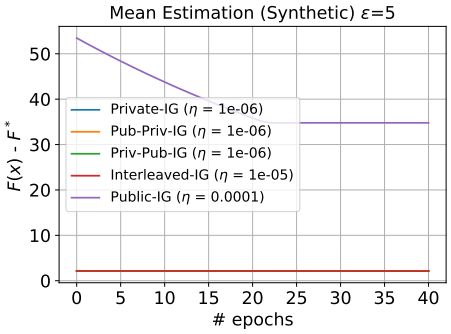

Figure 4.1: Using the public dataset completely is a bad idea.

Discussion. Figure 4.1 demonstrates that using only public samples leads to quick convergence to the mean of the surrogate dataset Ψ. However, there is a significant gap between the mean of the surrogate dataset and the mean of the true dataset. This highlights the necessity of incorporating private samples to achieve better convergence to the minimum of the true objective.

Figure 4.2 compares the convergence rates of various algorithms under a high privacy regime ($\epsilon = 5$, left) and a low privacy regime ($\epsilon = 10$, right). Interleaved-IG achieves the best convergence in the high privacy regime, while Private-IG performs best in the low privacy regime. Intuitively, in a high privacy regime, the noise added

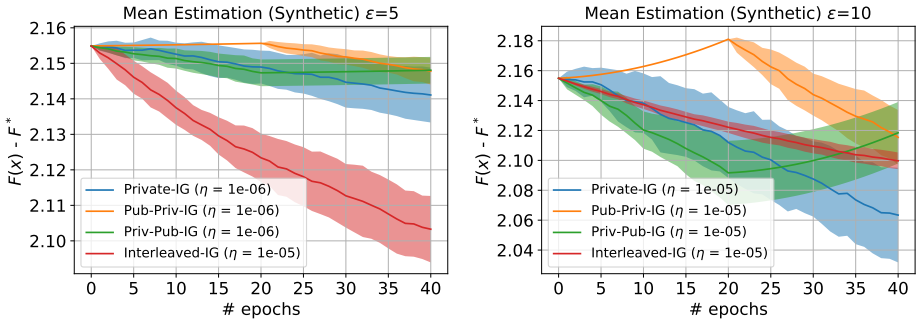

Figure 4.2: Large ϵ, smaller σ^2, all private becomes better than interleaved. Small ϵ, σ^2 large to use all private data. Interleaved is the best.

in Private-IG is substantial enough to hinder convergence significantly, making the effective use of public samples more critical.

The plot also shows that Interleaved-IG is better at making use of public samples to maximize the privacy-convergence trade-off, compared to Priv-Pub-IG and Pub-Priv-IG, as the theory indicates. This is because Interleaved-IG is able to use public samples to amplify privacy within each epoch based on PABI.

4.7 Conclusion

In this work, we investigate a private version of the Incremental Gradient (IG) method, the simplest form of shuffled gradient methods, taking one step towards bridging the gap between theory and practice in private optimization. We give the first comprehensive analysis from both optimization and privacy perspectives, with a particular focus on leveraging Privacy Amplification by Iteration (PABI) in the privacy analysis, which is under-explored in private optimization.

We introduce the Generalized IG Framework, which enables optimization using surrogate objectives and noise injection for privacy purpose. To analyze this framework, we propose a new dissimilarity metric to measure the difference between the surrogate and the true objective, specifically tailored to IG. We give the first empirical excess risk bound of DP-IG and compare it against that of DP-GD and DP-SGD. We further show that an interleaved training of using private and public samples in private IG, which we call Interleaved-IG, is able to better utilize public samples to

maximize the privacy-convergence trade-off. Finally, we empirically demonstrate the effectiveness of Interleaved-IG in a low privacy regime in the task of mean estimation, compared to other alternatives, including DP-IG.

Chapter 5

Proposed Work and Timeline

We propose two directions for extending the work in Chapter 4 on the private Incremental Gradient (IG) method, which is the most basic form of shuffled gradient methods.

5.1 Proposed Work 1: Generalized Shuffled Gradient Methods

Recall that as previously mentioned in Chapter 4, shuffled gradient methods come with three subtypes:

1. **Incremental Gradient (IG) Methods**: The order of samples is fixed at the start of training.
2. **Shuffle Once (SO)**: Samples are shuffled once before training begins.
3. **Random Reshuffling (RR)**: Samples are reshuffled at the beginning of each epoch.

While our *Generalized IG Framework* in Chapter 4 enables convergence analysis for IG with noise injection and surrogate objective functions, SO and RR are more commonly used in practice [79], for solving empirical risk minimization problems with a broad range of applications in privacy preserving machine learning. However, their private counterparts remain largely unexplored, similar to IG before our work.

To address this gap, we propose extending the *Generalized IG Framework* into a *Generalized Shuffled Gradient Framework* capable of analyzing SO and RR in both private and non-private settings. This extension involves three key objectives:

1. **Convergence Analysis.** Extend the proposed dissimilarity metric, which quantifies the difference between the true and surrogate objectives, from IG to SO and RR. Use this generalized framework to analyze the convergence of shuffled gradient methods across all subtypes.
2. **Empirical Excess Risk for Private SO and RR.** Develop private versions of SO and RR and derive empirical excess risk bounds for these methods. Compare these bounds against lower bounds and existing results for DP-GD and DP-SGD.
3. **Effective Use of Public Data.** Investigate strategies for leveraging public data samples in private SO and RR to maximize privacy-utility trade-offs. Develop methods that interleave private and public data effectively within the *Generalized Shuffled Gradient Framework*.

5.2 Proposed Work 2: Distributed / Decentralized Shuffled Gradient Methods

The current *Generalized IG Framework* and the proposed *Generalized Shuffled Gradient Framework* focus on analyzing shuffled gradient methods on a single machine. However, many real-world applications involve distributed or decentralized settings where data resides on edge clients, and the goal is collaborative training of a global model [57].

Figure 5.1 shows the difference between a distributed and a decentralized setting. In a distributed setting, see e.g., [57], there is one central server and a set of clients. In a decentralized setting, see e.g., [59], there is no central server while there is a communication graph that connects all clients. Each client can only interact with its neighbors connected by edges in the communication graph.

Although distributed and decentralized SGD have been extensively studied in both non-private [55, 59] and private settings [25, 115], shuffled gradient methods remain largely unexplored in these contexts. This gap in terms of convergence is particularly

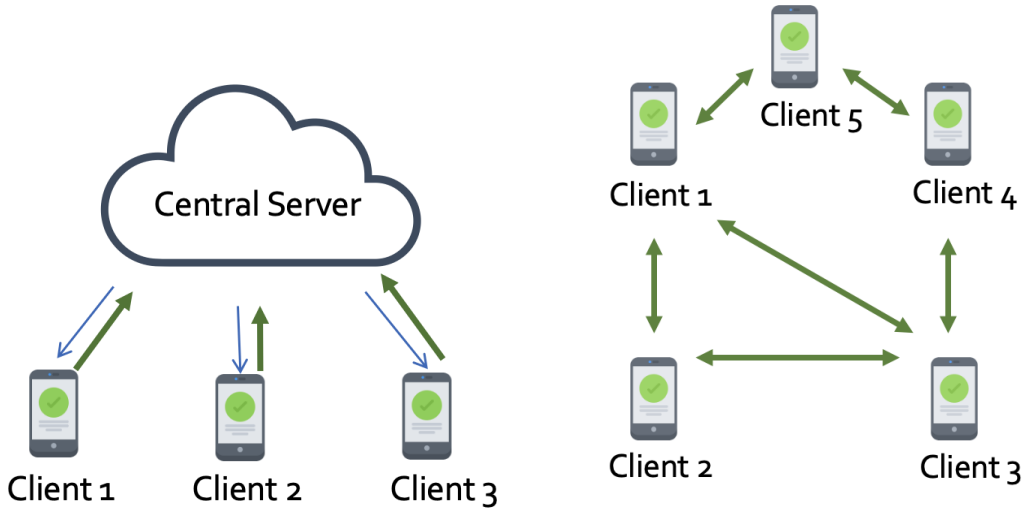

Figure 5.1: An illustration of distributed (left) and decentralized (right) settings.

intriguing because shuffled gradient methods, even in single-machine settings, exhibit a different convergence rate compared to SGD. Specifically, they achieve a convergence rate of $\mathcal{O}((\frac{1}{K})^{2/3})$ [69], where K is the number of epochs, compared to the $O(\frac{1}{\sqrt{T}})$ rate of SGD, where $T = nK$ is the total number of gradient steps. Extending this comparison to distributed and decentralized environments could reveal novel insights.

Moreover, we aim to explore how the following factors influence the convergence of distributed and decentralized shuffled gradient methods. These factors are well explored in distributed and decentralized SGD.

1. **Communication Efficiency.** Communication costs are a significant bottleneck in distributed and decentralized optimization. To address this, clients often perform $\tau > 1$ local gradient steps before exchanging updates. We aim to understand how the frequency of communication (τ) impacts the convergence bounds of shuffled gradient methods in these settings.
2. **Differential Privacy.** Privacy-preserving mechanisms, such as user-level differential privacy, are crucial in distributed and decentralized training to ensure that an adversary cannot infer the participation of a specific client. Additionally, [25] introduces a graph-dependent privacy notion in decentralized settings, where the communication topology affects privacy guarantees. We aim

to analyze how these achieving these privacy guarantees would influence the convergence rates of shuffled gradient methods.

3. **Client Data Heterogeneity.** Data heterogeneity—clients having non-identical data distributions—is a well-known challenge in Federated Learning and distributed SGD. Understanding its impact on distributed and decentralized shuffled gradient methods is essential for designing robust algorithms.

The goal of this work is to bridge the gap between single-machine and distributed/decentralized analyses, offering a comprehensive understanding of both private and non-private shuffled gradient methods in collaborative optimization scenarios.

5.3 Timeline

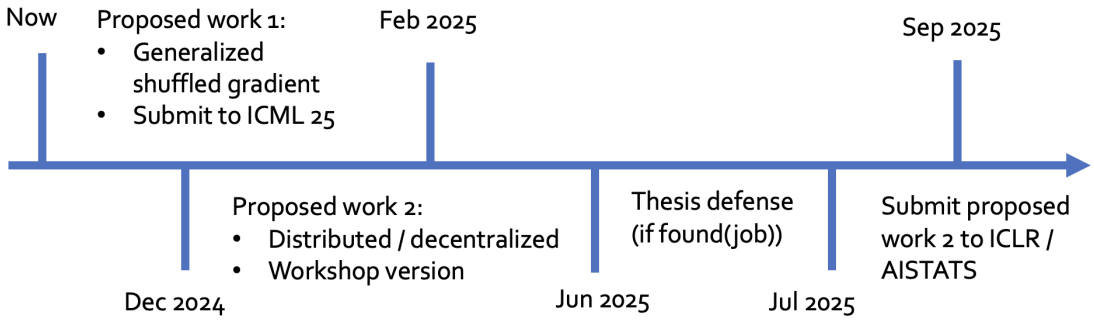

Figure 5.2: Timeline.

Appendix A

Correlated Distributed Mean Estimation

A.1 Additional Details on Motivation in Introduction

A.1.1 Preprocssing all client vectors by the same random matrix does not improve performance

Consider n clients. Suppose client i holds a vector $\mathbf{x}_i \in \mathbb{R}^d$. We want to apply Rand-$k$ or Rand-k-Spatial, while also making the encoding process more flexible than just randomly choosing k out of d coordinates. One naïve way of doing this is for each client to pre-process its vector by applying an orthogonal matrix $\boldsymbol{G} \in \mathbb{R}^{d \times d}$ that is the *same* across all clients. Such a technique might be helpful in improving the performance of quantization because the MSE due to quantization often depends on how uniform the coordinates of $\mathbf{x}_i$'s are, i.e. whether the coordinates of $\mathbf{x}_i$ have values close to each other. $\boldsymbol{G}$ is designed to be the random matrix (e.g. SRHT) that rotates $\mathbf{x}_i$ and makes its coordinates uniform.

Each client sends the server $\widehat{\mathbf{x}}_i = \boldsymbol{E}_i \boldsymbol{G} \mathbf{x}_i$, where $\boldsymbol{E}_i \in \mathbb{R}^{k \times d}$ is the subsamaping matrix. If we use Rand-k, the server can decode each client vector by first applying the decoding procedure of Rand-k and then rotating it back to the original space,

i.e., $\widehat{\mathbf{x}}_i^{(\text{Naïve})} = \frac{d}{k}\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i$. Note that

$$\begin{aligned}\mathbb{E}[\widehat{\mathbf{x}}_i^{(\text{Naïve})}] &= \frac{d}{k}\mathbb{E}[\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i] \\ &= \frac{d}{k}\boldsymbol{G}^T\frac{k}{d}\boldsymbol{I}_d\boldsymbol{G}\mathbf{x}_i \\ &= \mathbf{x}_i.\end{aligned}$$

Hence, $\widehat{\mathbf{x}}_i^{(\text{Naïve})}$ is unbiased. The MSE of $\widehat{\mathbf{x}}^{(\text{Naïve})} = \frac{1}{n}\sum_{i=1}^n \widehat{\mathbf{x}}_i^{(\text{Naïve})}$ is given as

$$\begin{aligned}&\mathbb{E}\left\|\bar{\mathbf{x}} - \widehat{\mathbf{x}}^{(\text{Naïve})}\right\|_2^2 = \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i - \frac{1}{n}\frac{d}{k}\sum_{i=1}^n \boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i\right\|_2^2 \\ &= \frac{1}{n^2}\mathbb{E}\left\|\sum_{i=1}^n \mathbf{x}_i - \frac{d}{k}\sum_{i=1}^n \boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i\right\|_2^2 \\ &= \frac{1}{n^2}\left\{\frac{d^2}{k^2}\mathbb{E}\left\|\sum_{i=1}^n \boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i\right\|^2 - \left\|\sum_{i=1}^n \mathbf{x}_i\right\|^2\right\} \\ &= \frac{1}{n^2}\left\{\frac{d^2}{k^2}\left(\sum_{i=1}^n \mathbb{E}\|\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i\|_2^2 + \sum_{i\neq j}\mathbb{E}\left\langle\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i, \boldsymbol{G}\boldsymbol{E}_l^T\boldsymbol{E}_l\boldsymbol{G}\mathbf{x}_l\right\rangle\right) - \left\|\sum_{i=1}^n \mathbf{x}_i\right\|^2\right\}.\end{aligned} \tag{A.1}$$

Next, we bound the first term in Eq. A.1.

$$\begin{aligned}&\mathbb{E}\|\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i\|_2^2 = \mathbb{E}[\mathbf{x}_i^T\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i] = \mathbb{E}[\mathbf{x}_i^T\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i] \\ &= \mathbf{x}_i^T\boldsymbol{G}^T\mathbb{E}[(\boldsymbol{E}_i^T\boldsymbol{E}_i)^2]\boldsymbol{G}\mathbf{x}_i \\ &= \mathbf{x}_i^T\frac{k}{d}\boldsymbol{I}_d\mathbf{x}_i \qquad (\because (\boldsymbol{E}_i^T\boldsymbol{E}_i)^2 = \boldsymbol{E}_i^T\boldsymbol{E}_i) \\ &= \frac{k}{d}\|\mathbf{x}_i\|_2^2 \end{aligned} \tag{A.2}$$

The second term in Eq. A.1 can also be simplified as follows.

$$\begin{aligned}&\mathbb{E}[\langle\boldsymbol{G}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{G}\mathbf{x}_i, \boldsymbol{G}^T\boldsymbol{E}_l^T\boldsymbol{E}_l\boldsymbol{G}\mathbf{x}_l\rangle] \\ &= \langle\boldsymbol{G}^T\mathbb{E}[\boldsymbol{E}_i^T\boldsymbol{E}_i]\boldsymbol{G}\mathbf{x}_i, \boldsymbol{G}^T\mathbb{E}[\boldsymbol{E}_l^T\boldsymbol{E}_l]\boldsymbol{G}\mathbf{x}_l\rangle\end{aligned}$$

$$= \langle \boldsymbol{G}^T \frac{k}{d} \boldsymbol{I}_d \boldsymbol{G} \mathbf{x}_i, \boldsymbol{G}^T \frac{k}{d} \boldsymbol{I}_d \boldsymbol{G} \mathbf{x}_l \rangle$$
$$= \frac{k^2}{d^2} \langle \mathbf{x}_i, \mathbf{x}_l \rangle. \tag{A.3}$$

Plugging Eq. A.2 and Eq. A.3 into Eq. A.1, we get the MSE is

$$\begin{aligned}
&\mathbb{E}\|\bar{\mathbf{x}} - \widehat{\mathbf{x}}^{(\text{Naïve})}\|_2^2 \\
&= \frac{1}{n^2}\Big\{\frac{d^2}{k^2}\Big(\sum_{i=1}^n \frac{k}{d}\|\mathbf{x}_i\|_2^2 + 2\sum_{i=1}^n \sum_{l=i+1}^n \frac{k^2}{d^2}\langle \mathbf{x}_i, \mathbf{x}_l\rangle\Big) - \Big\|\sum_{i=1}^n \mathbf{x}_i\Big\|^2\Big\} \\
&= \frac{1}{n^2}(\frac{d}{k} - 1)\sum_{i=1}^n \|\mathbf{x}_i\|_2^2,
\end{aligned}$$

which has exactly the same MSE as that of Rand-k. The problem is that if each client applies the same rotational matrix $\boldsymbol{G}$, simply rotating the vectors will not change the ℓ_2 norm of the decoded vector, and hence the MSE. Similarly, if one applies Rand-k-Spatial, one ends up having exactly the same MSE as that of Rand-k-Spatial as well. Hence, we need to design a new decoding procedure when the encoding procedure at the clients are more flexible.

A.1.2 $nk \gg d$ is not interesting

One can rewrite $\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i$ in the Rand-Proj-Spatial estimator (Eq. 2.5) as $\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i = \sum_{j=1}^{nk} \mathbf{g}_i \mathbf{g}_i^T$, where $\mathbf{g}_j \in \mathbb{R}^d$ and $\mathbf{g}_{ik}, \mathbf{g}_{ik+1}, \ldots, \mathbf{g}_{(i+1)k}$ are the rows of $\boldsymbol{G}_i$. Since when $nk \gg d$, $\sum_{j=1}^{nk} \mathbf{g}_j \mathbf{g}_j^T \to \mathbb{E}[\sum_{j=1}^n \mathbf{g}_j \mathbf{g}_j^T]$ due to Law of Large Numbers, one way to see the limiting MSE of Rand-Proj-Spatial when nk is large is to approximate $\sum_{i=1}^n \sum_{j=1}^{nk} \mathbf{g}_i \mathbf{g}_i^T$ by its expectation.

By Lemma 2.4.1, when $\boldsymbol{G}_i = \boldsymbol{E}_i$, Rand-Proj-Spatial recovers Rand-k-Spatial. We now discuss the limiting behavior of Rand-k-Spatial when $nk \gg d$ by leveraging our proposed Rand-Proj-Spatial. In this case, each $\mathbf{g}_j$ can be viewed as a random based vector $\mathbf{e}_w$ for w randomly chosen in $[d]$. $\sum_{i=1}^{nk} \mathbf{g}_j \mathbf{g}_j^T \to \mathbb{E}[\sum_{i=1}^{nk} \mathbf{g}_j \mathbf{g}_j^T] = \sum_{i=1}^{nk} \frac{1}{d}\boldsymbol{I}_d = \frac{nk}{d}\boldsymbol{I}_d$. And so the scalar $\bar{\beta}$ in Eq. 2.5 to ensure an unbiased estimator is computed as

$$\bar{\beta}\mathbb{E}[(\frac{nk}{d}\boldsymbol{I}_d)^\dagger \boldsymbol{G}_i^T \boldsymbol{G}_i] = \boldsymbol{I}_d$$

$$\bar{\beta}\frac{d}{nk}\boldsymbol{I}_d\mathbb{E}[\boldsymbol{G}_i^T\boldsymbol{G}_i] = \boldsymbol{I}_d$$
$$\bar{\beta}\frac{d}{nk}\frac{k}{d} = \boldsymbol{I}_d$$
$$\bar{\beta} = n$$

And the MSE is now

$$\mathbb{E}\Big[\|\bar{\mathbf{x}} - \hat{\mathbf{x}}\|\Big] = \mathbb{E}\Big[\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i - \frac{1}{n}\bar{\beta}\frac{d}{nk}\boldsymbol{I}_d\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i\|_2^2\Big]$$
$$= \frac{1}{n^2}\Big\{\bar{\beta}^2\frac{d^2}{n^2k^2}\mathbb{E}\Big[\|\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i\|_2^2\Big] - \|\sum_{i=1}^n \mathbf{x}_i\|_2^2\Big\}$$
$$= \frac{1}{n^2}\Big\{n^2\frac{d^2}{n^2k^2}\Big(\sum_{i=1}^n \mathbb{E}\Big[\|\boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i\|_2^2\Big] + 2\sum_{i=1}^n\sum_{l=i+1}^n \langle \boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i, \boldsymbol{E}_l^T\boldsymbol{E}_l\mathbf{x}_l\rangle\Big]\Big) - \|\sum_{i=1}^n \mathbf{x}_i\|_2^2\Big\}$$
$$= \frac{1}{n^2}\Big\{\frac{d^2}{k^2}\Big(\sum_{i=1}^n \mathbb{E}\Big[\mathbf{x}_i^T(\boldsymbol{E}_i^T\boldsymbol{E}_i)^2\mathbf{x}_i\Big] + 2\sum_{i=1}^n\sum_{l=i+1}^n \frac{k^2}{d^2}\langle \mathbf{x}_i, \mathbf{x}_l\rangle\Big) - \|\sum_{i=1}^n \mathbf{x}_i\|_2^2\Big\}$$
$$= \frac{1}{n^2}\Big\{\frac{d^2}{k^2}\Big(\sum_{i=1}^n \frac{k}{d}\|\mathbf{x}_i\|_2^2 + 2\sum_{i=1}^n\sum_{l=i+1}^n \frac{k^2}{d^2}\langle \mathbf{x}_i, \mathbf{x}_l\rangle\Big) - \sum_{i=1}^n \|\mathbf{x}_i\|_2^2 - 2\sum_{i=1}^n\sum_{l=i+1}^n \langle \mathbf{x}_i, \mathbf{x}_l\rangle\Big\}$$
$$= \frac{1}{n^2}(\frac{d}{k} - 1)\sum_{i=1}^n \|\mathbf{x}_i\|_2^2$$

which is exactly the same MSE as Rand-k. This implies when nk is large, the MSE of Rand-k-Spatial does not get improved compared to Rand-k with correlation information. Intuitively, this implies when $nk \gg d$, the server gets enough amount of information from the client, and does not need correlation to improve its estimator. Hence, we focus on the more interesting case when $nk < d$ — that is, when the server does not have enough information from the clients, and thus wants to use additional information, i.e. cross-client correlation, to improve its estimator.

A.2 Additional Details on the Rand-Proj-Spatial Family Estimator

A.2.1 $\bar{\beta}$ is a scalar

From Eq. A.9 in the proof of Theorem 2.4.3 and Eq. A.14 in the proof of Theorem 2.4.4, it is evident that the unbiasedness of the mean estimator $\widehat{\mathbf{x}}^{\text{Rand-Proj-Spatial}}$ is ensured collectively by

- The random sampling matrices $\{\boldsymbol{E}_i\}$.
- The orthogonality of scaled Hadamard matrices $\boldsymbol{H}^T\boldsymbol{H} = d\boldsymbol{I}_d = \boldsymbol{H}\boldsymbol{H}^T$.
- The rademacher diagonal matrices, with the property $(\boldsymbol{D}_i)^2 = \boldsymbol{I}_d$.

A.2.2 Alternative motivating regression problems

Alternative motivating regression problem 1.

Let $\boldsymbol{G}_i \in \mathbb{R}^{k\times d}$ and $\boldsymbol{W}_i \in \mathbb{R}^{d\times k}$ be the encoding and decoding matrix for client i. One possible alternative estimator that translates the intuition that the decoded vector should be close to the client's original vector, for all clients, is by solving the following regression problem,

$$\begin{aligned}\hat{\mathbf{x}} &= \underset{\boldsymbol{W}}{\operatorname{argmin}}\, f(\boldsymbol{W}) = \mathbb{E}[\|\bar{\mathbf{x}} - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i\|_2^2] \\ &\text{subject to } \bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i] \end{aligned} \tag{A.4}$$

where $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_n)$ and the constraint enforces unbiasedness of the estimator. The estimator is then the solution of the above problem. However, we note that optimizing a decoding matrix $\boldsymbol{W}_i$ for each client leads to performing individual decoding of each client's compressed vector instead of a joint decoding process that considers all clients' compressed vectors. Only a joint decoding process can achieve the goal of leveraging cross-client information to reduce the estimation error. Indeed, we show as follows that solving the above optimization problem in Eq. A.4 recovers the MSE of our baseline Rand-k. Note

$$
\begin{aligned}
f(\boldsymbol{W}) &= \mathbb{E}[\|\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i)\|_2^2] = \mathbb{E}[\|\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i\|_2^2] \\
&= \mathbb{E}\Big[\frac{1}{n^2}\Big(\sum_{i=1}^{n}\|(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i)\|_2^2 + \sum_{i\neq j}\Big\langle(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i, (\boldsymbol{I}_d - \boldsymbol{W}_j\boldsymbol{G}_j)\mathbf{x}_j\Big\rangle\Big)\Big] \\
&= \frac{1}{n^2}\Big(\sum_{i=1}^{n}\mathbb{E}\Big[\|(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i\|_2^2\Big] + \sum_{i\neq j}\mathbb{E}\Big[\Big\langle(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i, (\boldsymbol{I}_d - \boldsymbol{W}_j\boldsymbol{G}_j)\mathbf{x}_j\Big\rangle\Big]\Big).
\end{aligned}
\tag{A.5}
$$

By the constraint of unbiasedness, i.e., $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i]$, there is

$$
\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i] = 0 \Leftrightarrow \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i] = 0.
$$

We now show that a sufficient and necessary condition to satisfy the above unbiasedness constraint is that for all $i \in [n]$, $\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] = \boldsymbol{I}_d$.

Sufficiency. It is obvious that if for all $i \in [n]$, $\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] = \boldsymbol{I}_d$, then we have $\frac{1}{n}\mathbb{E}[(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i] = 0$.

Necessity. Consider the special case that for some $i \in [n]$ and $\lambda \in [d]$, $\mathbf{x}_i = n\mathbf{e}_\lambda$, where $\mathbf{e}_\lambda$ is the λ-th canonical basis vector, and $\mathbf{x}_j = 0$, and for all $j \in [n] \setminus \{i\}$. Then,

$$
\mathbf{e}_\lambda = \bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i] = \frac{1}{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i]\mathbf{e}_\lambda = [\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i]]_\lambda,
$$

where $[\cdot]_\lambda$ denotes the λ-th column of matrix $\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i]$.

Since our approach is agnostic to the choice of vectors, we need this choice of decoder matrices, by varying λ over $[d]$, we see that we need $\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] = \boldsymbol{I}_d$. And by varying i over $[n]$, we see that we need $\mathbb{E}[\boldsymbol{W}_j\boldsymbol{G}_j] = \boldsymbol{I}_d$ for all $j \in [n]$.

Therefore, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i] \Leftrightarrow \forall i \in [n], \mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] = \boldsymbol{I}_d$.

This implies the second term of $f(\boldsymbol{W})$ in Eq. A.5 is 0, that is,

$$
\sum_{i\neq j}\mathbb{E}\Big[\Big\langle(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i, (\boldsymbol{I}_d - \boldsymbol{W}_j\boldsymbol{G}_j)\mathbf{x}_j\Big\rangle = 0.
$$

Hence, we only need to solve

$$\hat{\mathbf{x}} = \underset{\boldsymbol{W}}{\operatorname{argmin}} f_2(\boldsymbol{W}) = \sum_{i=1}^{n} \mathbb{E}\Big[\|(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i\|_2^2\Big] \tag{A.6}$$

Since each $\boldsymbol{W}_i$ appears in $f_2(\boldsymbol{W})$ separately, each $\boldsymbol{W}_i$ can be optimized separately, via solving

$$\min_{\boldsymbol{W}_i} \mathbb{E}\Big[\|(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i\|_2^2\Big] \quad \text{subject to } \mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] = \boldsymbol{I}_d.$$

One natural solution is to take $\boldsymbol{W}_i = \frac{d}{k}\boldsymbol{G}_i^{\dagger}$, $\forall i \in [n]$. For $i \in [n]$, let $\boldsymbol{G}_i = \boldsymbol{V}_i\Lambda_i\boldsymbol{U}_i^T$ be its SVD, where $\boldsymbol{V}_i \in \mathbb{R}^{k\times d}$ and $\boldsymbol{U}_i \in \mathbb{R}^{d\times d}$ are orthogonal matrices. Then,

$$\boldsymbol{W}_i\boldsymbol{G}_i = \frac{d}{k}\boldsymbol{U}_i\Lambda_i^{\dagger}\boldsymbol{V}_i^T\boldsymbol{V}_i\Lambda\boldsymbol{U}^T = \frac{d}{k}\boldsymbol{U}_i\Lambda_i^{\dagger}\Lambda\boldsymbol{U}^T = \frac{d}{k}\boldsymbol{U}_i\Sigma\boldsymbol{U}_i^T,$$

where Σ is a diagonal matrix with 0s and 1s on the diagonal.

For simplicity, we assume the random matrix $\boldsymbol{U}_i$ follows a continuous distribution. $\boldsymbol{U}_i$ being discrete follows a similar analysis. Let $\mu(\boldsymbol{U}_i)$ be the measure of $\boldsymbol{U}_i$.

$$\begin{aligned}
\mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i] &= \frac{d}{k}\mathbb{E}[\boldsymbol{U}_i\Sigma\boldsymbol{U}_i^T] = \frac{d}{k}\int_{\boldsymbol{U}_i}\mathbb{E}[\boldsymbol{U}_i\Sigma_i\boldsymbol{U}_i^T \mid \boldsymbol{U}_i]\cdot d\mu(\boldsymbol{U}_i)\\
&= \frac{d}{k}\int_{\boldsymbol{U}_i}\boldsymbol{U}_i\mathbb{E}[\Sigma_i \mid \boldsymbol{U}_i]\boldsymbol{U}_i^T\cdot\mu(\boldsymbol{U}_i)\\
&= \frac{d}{k}\int_{\boldsymbol{U}_i}\boldsymbol{U}_i\frac{k}{d}\boldsymbol{I}_d\boldsymbol{U}_i^T\cdot d\mu(\boldsymbol{U}_i)\\
&= \frac{d}{k}\frac{k}{d}\boldsymbol{I}_d = \boldsymbol{I}_d,
\end{aligned}$$

which means the estimator $\frac{1}{n}\sum_{i=1}^{n}\frac{k}{d}\boldsymbol{G}_i^{\dagger}\boldsymbol{G}_i$ satisfies unbiasedness. The MSE is now

$$\begin{aligned}
MSE &= \mathbb{E}\Big[\|\bar{\mathbf{x}} - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i\|_2^2\Big] = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\Big[\|(\boldsymbol{I}_d - \boldsymbol{W}_i\boldsymbol{G}_i)\mathbf{x}_i\|_2^2\Big]\\
&= \frac{1}{n^2}\sum_{i=1}^{n}\Big(\|\mathbf{x}_i\|_2^2 + \mathbb{E}[\|\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i\|_2^2] - 2\langle\mathbf{x}_i, \mathbb{E}[\boldsymbol{W}_i\boldsymbol{G}_i]\mathbf{x}_i\rangle\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n^2}\sum_{i=1}^{n}\Big(\|\mathbf{x}_i\|_2^2 + \mathbb{E}[\|\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i\|_2^2] - 2\langle \mathbf{x}_i, \mathbf{x}_i\rangle\Big)\\
&= \frac{1}{n^2}\sum_{i=1}^{n}\Big(\mathbb{E}[\|\boldsymbol{W}_i\boldsymbol{G}_i\mathbf{x}_i\|_2^2 - \|\mathbf{x}_i\|_2^2]\Big)\\
&= \frac{1}{n^2}\sum_{i=1}^{n}\Big(\mathbf{x}_i\mathbb{E}[(\boldsymbol{W}_i\boldsymbol{G}_i)^T(\boldsymbol{W}_i\boldsymbol{G}_i)]\mathbf{x}_i - \|\mathbf{x}_i\|_2^2\Big).
\end{aligned}
$$

Again, let $\boldsymbol{G}_i = \boldsymbol{V}_i\Lambda_i\boldsymbol{U}_i^T$ be its SVD and consider $\boldsymbol{W}_i\boldsymbol{G}_i = \frac{d}{k}\boldsymbol{U}_i\Sigma_i\boldsymbol{U}_i^T$, where Σ_i is a diagonal matrix with 0s and 1s. Then,

$$
\begin{aligned}
MSE &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i=1}^{n}\Big(\mathbf{x}_i^T\frac{d^2}{k^2}\mathbb{E}[\boldsymbol{U}_i\Sigma_i\boldsymbol{U}_i^T\boldsymbol{U}_i\Sigma_i\boldsymbol{U}_i^T]\mathbf{x}_i - \|\mathbf{x}_i\|_2^2\Big)\\
&= \frac{1}{n^2}\sum_{i=1}^{n}\Big(\frac{d^2}{k^2}\mathbf{x}_i^T\mathbb{E}[\boldsymbol{U}_i\Sigma^2\boldsymbol{U}_i^T]\mathbf{x}_i - \|\mathbf{x}_i\|_2^2\Big).
\end{aligned}
$$

Since $\boldsymbol{G}_i$ has rank k, Σ_i is a diagonal matrix with k out of d entries being 1 and the rest being 0. Let $\mu(\boldsymbol{U}_i)$ be the measure of $\boldsymbol{U}_i$. Hence, for $i \in [n]$,

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{U}_i\Sigma_i^2\boldsymbol{U}_i^T] &= \int_{\boldsymbol{U}_i}\mathbb{E}[\boldsymbol{U}_i\Sigma_i^2\boldsymbol{U}_i^T \mid \boldsymbol{U}_i]d\mu(\boldsymbol{U}_i)\\
&= \int_{\boldsymbol{U}_i}\boldsymbol{U}_i\mathbb{E}[\Sigma_i^2 \mid \boldsymbol{U}_i]\boldsymbol{U}_i^T d\mu(\boldsymbol{U}_i)\\
&= \int_{\boldsymbol{U}_i}\frac{k}{d}\boldsymbol{U}_i\boldsymbol{I}_d\boldsymbol{U}_i^T d\mu(\boldsymbol{U}_i)\\
&= \frac{k}{d}\int_{\boldsymbol{U}_i}\boldsymbol{I}_d d\mu(\boldsymbol{U}_i)\\
&= \frac{k}{d}\boldsymbol{I}_d.
\end{aligned}
$$

Therefore, the MSE of the estimator, which is the solution of the optimization problem in Eq. A.4, is

$$
MSE = \frac{1}{n^2}\sum_{i=1}^{n}\Big(\frac{d^2}{k^2}\mathbf{x}_i^T\frac{k}{d}\boldsymbol{I}_d\mathbf{x}_i - \|\mathbf{x}_i\|_2^2\Big) = \frac{1}{n^2}(\frac{d}{k} - 1)\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2,
$$

which is the same MSE as that of Rand-k.

Alternative motivating regression problem 2.

Another motivating regression problem based on which we can design our estimator is

$$\widehat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\frac{1}{n}\sum_{i=1}^{n} \mathbf{G}_i\mathbf{x} - \frac{1}{n}\sum_{i=1}^{n} \mathbf{G}_i\mathbf{x}_i\|_2^2 \tag{A.7}$$

Note that $\boldsymbol{G}_i \in \mathbb{R}^{k\times d}, \forall i \in [n]$, and so the solution to the above problem is

$$\widehat{\mathbf{x}}^{(\text{solution})} = \Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\Big)^{\dagger}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\mathbf{x}_i\Big),$$

and to ensure unbiasedness of the estimator, we can set $\bar{\beta} \in \mathbb{R}$ and have the estimator as

$$\widehat{\mathbf{x}}^{(\text{estimator})} = \bar{\beta}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\Big)^{\dagger}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\mathbf{x}_i\Big).$$

It is not hard to see this estimator does not lead to an MSE as low as Rand-Proj-Spatial does. Consider the full correlation case, i.e., $\mathbf{x}_i = \mathbf{x}, \forall i \in [n]$, for example, the estimator is now

$$\widehat{\mathbf{x}}^{(\text{estimator})} = \bar{\beta}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\Big)^{\dagger}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i\Big)\mathbf{x}.$$

Note that $\operatorname{rank}(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}_i)$ is at most k, since $\boldsymbol{G}_i \in \mathbb{R}^{k\times d}$, $\forall i \in [k]$. This limits the amount of information of $\mathbf{x}$ the server can recover.

While recall that in this case, the Rand-Proj-Spatial estimator is

$$\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial})} = \bar{\beta}\Big(\sum_{i=1}^{n} \boldsymbol{G}_i^T\boldsymbol{G}_i\Big)^{\dagger}\sum_{i=1}^{n} \boldsymbol{G}_i^T\boldsymbol{G}_i\mathbf{x} = \bar{\beta}\boldsymbol{S}^{\dagger}\boldsymbol{S}\mathbf{x},$$

where $\boldsymbol{S}$ can have rank at most nk.

A.2.3 Why deriving the MSE of Rand-Proj-Spatial with SRHT is hard

To analyze Eq. 2.11, one needs to compute the distribution of eigendecomposition of $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$, i.e. the sum of the covariance of SRHT. To the best of our knowledge, there is no non-trivial closed form expression of the distribution of eigen-decomposition of even a single $\boldsymbol{G}_i^T \boldsymbol{G}_i$, when $\boldsymbol{G}_i$ is SRHT, or other commonly used random matrices, e.g. Gaussian. When $\boldsymbol{G}_i$ is SRHT, since $\boldsymbol{G}_i^T \boldsymbol{G}_i = \boldsymbol{D}_i \boldsymbol{H} \boldsymbol{E}_i^T \boldsymbol{E}_i \boldsymbol{H} \boldsymbol{D}_i$ and the eigenvalues of $\boldsymbol{E}_i^T \boldsymbol{E}_i$ are just diagonal entries, one might attempt to analyze $\boldsymbol{H}\boldsymbol{D}_i$. While the hardmard matrix $\boldsymbol{H}$'s eigenvalues and eigenvectors are known[1], the result can hardly be applied to analyze the distribution of singular values or singular vectors of $\boldsymbol{H}\boldsymbol{D}_i$.

Even if one knows the eigen-decomposition of a single $\boldsymbol{G}_i^T \boldsymbol{G}_i$, it is still hard to get the eigen-decomposition of $\boldsymbol{S}$. The eigenvalues of a matrix $\boldsymbol{A}$ can be viewed as a non-linear function in the $\boldsymbol{A}$, and hence it is in general hard to derive closed form expressions for the eigenvalues of $\boldsymbol{A} + \boldsymbol{B}$, given the eigenvalues of $\boldsymbol{A}$ and that of $\boldsymbol{B}$. One exception is when $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same eigenvector and the eigenvalues of $\boldsymbol{A} + \boldsymbol{B}$ becomes a sum of the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$. Recall when $\boldsymbol{G}_i = \boldsymbol{E}_i$, Rand-Proj-Spatial recovers Rand-k-Spatial. Since $\boldsymbol{E}_i^T \boldsymbol{E}_i$'s all have the same eigenvectors (i.e. same as $\boldsymbol{I}_d$), the eigenvalues of $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{E}_i^T \boldsymbol{E}_i$ are just the sum of diagonal entries of $\boldsymbol{E}_i^T \boldsymbol{E}_i$'s. Hence, deriving the MSE for Rand-k-Spatial is not hard compared to the more general case when $\boldsymbol{G}_i^T \boldsymbol{G}_i$'s can have different eigenvectors.

Since one can also view $\boldsymbol{S} = \sum_{i=1}^{nk} \mathbf{g}_i \mathbf{g}_i^T$, i.e. the sum of nk rank-one matrices, one might attempt to recursively analyze the eigen-decomposition of $\sum_{i=1}^{n'} \mathbf{g}_i \mathbf{g}_i^T + \mathbf{g}_{n'+1} \mathbf{g}_{n'+1}^T$ for $n' \leq n$. One related problem is eigen-decomposition of a low-rank updated matrix in perturbation analysis: Given the eigen-decomposition of a matrix $\boldsymbol{A}$, what is the eigen-decomposition of $\boldsymbol{A} + \boldsymbol{V}\boldsymbol{V}^T$, where $\boldsymbol{V}$ is low-rank matrix (or more commonly rank-one)? To compute the eigenvalues of $\boldsymbol{A} + \boldsymbol{V}\boldsymbol{V}^T$ directly from that of $\boldsymbol{A}$, the most effective and widely applied solution is to solve the so-called secular equation, e.g. [8, 43, 45]. While this can be done computationally efficiently, it is hard to get a closed form expression for the eigenvalues of $\boldsymbol{A} + \boldsymbol{V}\boldsymbol{V}^T$ from the secular equation.

[1] See this note https://core.ac.uk/download/pdf/81967428.pdf

The previous analysis of SRHT in e.g. [4, 63, 64, 65, 105] is based on asymptotic properties of SRHT, such as the limiting eigen-spectrum, or concentration bounds that bounds the singular values. To analyze the MSE of Rand-Proj-Spatial, however, we need an exact, non-asymptotic analysis of the distribution of SRHT. Concentration bounds does not apply, since computing the pseudo-inverse in Eq. 2.5 naturally bounds the eigenvalues, and applying concentration bounds will only lead to a loose upper bound on MSE.

A.2.4 More simulation results on incorporating various degrees of correlation

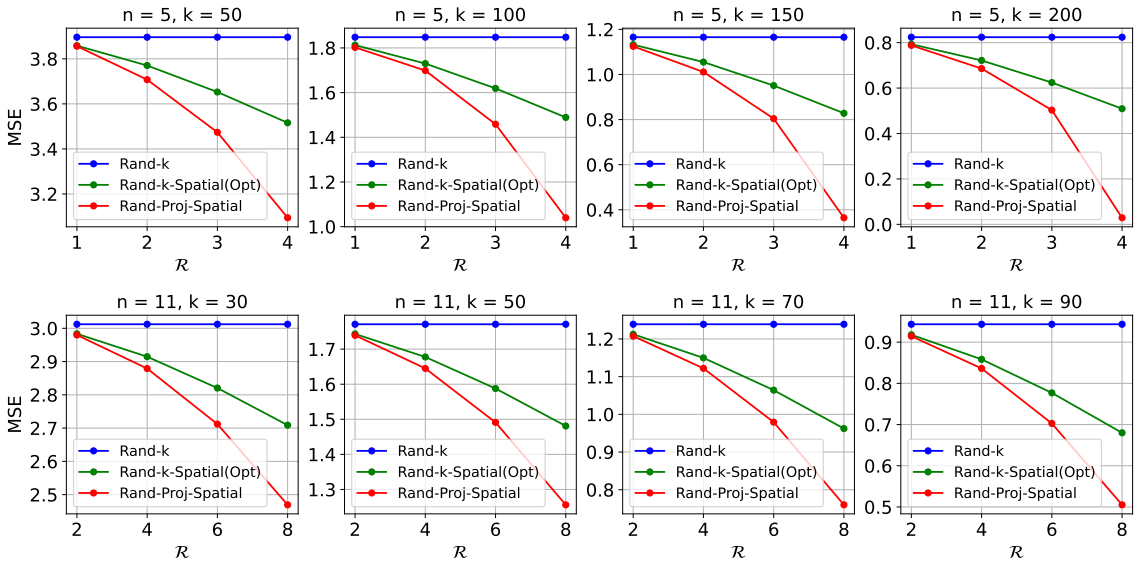

Figure A.1: MSE comparison of estimators Rand-k, Rand-k-Spatial(Opt), Rand-Proj-Spatial, given the degree of correlation $\mathcal{R}$. Rand-k-Spatial(Opt) denotes the estimator that gives the lowest possible MSE from the Rand-k-Spatial family. We consider $d = 1024$, a smaller number of clients $n \in \{5, 11\}$, and k values such that $nk < d$. In each plot, we fix n, k, d and vary the degree of positive correlation $\mathcal{R}$. Note the range of $\mathcal{R}$ is $\mathcal{R} \in [0, n-1]$. We choose $\mathcal{R}$ with equal space in this range.

A.3 All Proof Details

A.3.1 Proof of Theorem 2.4.3

Theorem 4.3 (MSE under Full Correlation)**.** *Consider n clients, each holding the same vector $\mathbf{x} \in \mathbb{R}^d$. Suppose we set $T(\lambda) = \lambda$, $\bar{\beta} = \frac{d}{k}$ in Eq. 2.5, and the random linear map $\boldsymbol{G}_i$ at each client to be an SRHT matrix. Let δ be the probability that $\boldsymbol{S} = \sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i$ does not have full rank. Then, for $nk \leq d$,*

$$\mathbb{E}\Big[\|\widehat{\mathbf{x}}^{(\textit{Rand-Proj-Spatial(Max)})} - \bar{\mathbf{x}}\|_2^2\Big] \leq \Big[\frac{d}{(1-\delta)nk + \delta k} - 1\Big]\|\mathbf{x}\|_2^2 \tag{A.8}$$

Proof. All clients have the same vector $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n = \mathbf{x} \in \mathbb{R}^d$. Hence, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i = \mathbf{x}$, and the decoding scheme is

$$\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial(Max)})} = \bar{\beta}\Big(\sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i\Big)^\dagger \sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i \mathbf{x} = \bar{\beta}\boldsymbol{S}^\dagger \boldsymbol{S}\mathbf{x},$$

where $\boldsymbol{S} = \sum_{i=1}^n \boldsymbol{G}_i^T \boldsymbol{G}_i$. Let $\boldsymbol{S} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$ be its eigendecomposition. Since $\boldsymbol{S}$ is a real symmetric matrix, $\boldsymbol{U}$ is orthogonal, i.e., $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_d = \boldsymbol{U}\boldsymbol{U}^T$. Also, $\boldsymbol{S}^\dagger = \boldsymbol{U}\Lambda^\dagger\boldsymbol{U}^T$, where $\Lambda^\dagger$ is a diagonal matrix, such that

$$[\Lambda^\dagger]_{ii} = \begin{cases} 1/[\Lambda]_{ii} & \text{if } \Lambda_{ii} \neq 0, \\ 0 & \text{else.} \end{cases}$$

Let δ_c be the probability that $\boldsymbol{S}$ has rank c, for $c \in \{k, k+1, \ldots, nk-1\}$. Note that $\delta = \sum_{c=k}^{nk-1} \delta_c$. For vector $\mathbf{m} \in \mathbb{R}^d$, we use $\text{diag}(\mathbf{m}) \in \mathbb{R}^{d\times d}$ to denote the matrix whose diagonal entries correspond to the coordinates of $\mathbf{m}$ and the rest of the entries are zeros.

Computing $\bar{\beta}$. First, we compute $\bar{\beta}$. To ensure that our estimator $\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial(Max)})}$ is unbiased, we need $\bar{\beta}\mathbb{E}[\boldsymbol{S}^\dagger\boldsymbol{S}\mathbf{x}] = \mathbf{x}$. Consequently,

$$\begin{aligned} \mathbf{x} &= \bar{\beta}\mathbb{E}[\boldsymbol{U}\Lambda^\dagger\boldsymbol{U}^T\boldsymbol{U}\Lambda\boldsymbol{U}^T]\mathbf{x} \\ &= \bar{\beta}\left[\sum_{\boldsymbol{U}=\Phi} \Pr[\boldsymbol{U} = \Phi]\mathbb{E}[\boldsymbol{U}\Lambda^\dagger\Lambda\boldsymbol{U}^T \mid \boldsymbol{U} = \Phi]\right]\mathbf{x} \end{aligned}$$

$$
\begin{aligned}
&= \bar{\beta}\left[\sum_{\boldsymbol{U}=\Phi} \Pr[\boldsymbol{U}=\Phi]\boldsymbol{U}\mathbb{E}[\Lambda^{\dagger}\Lambda \mid \boldsymbol{U}=\Phi]\boldsymbol{U}^T\right]\mathbf{x} \\
&\overset{(a)}{=} \bar{\beta}\left[\sum_{\boldsymbol{U}=\Phi} \Pr[\boldsymbol{U}=\Phi]\boldsymbol{U}\mathbb{E}[\mathrm{diag}(\mathbf{m}) \mid \boldsymbol{U}=\Phi]\boldsymbol{U}^T\right]\mathbf{x} \\
&\overset{(b)}{=} \bar{\beta}\sum_{\boldsymbol{U}=\Phi} \Pr[\boldsymbol{U}=\Phi]\left[\boldsymbol{U}\Big((1-\delta)\frac{nk}{d}\boldsymbol{I}_d + \sum_{c=k}^{nk-1}\delta_c\frac{c}{d}\boldsymbol{I}_d\Big)\boldsymbol{U}^T\right]\mathbf{x} \\
&= \bar{\beta}\Big[(1-\delta)\frac{nk}{d} + \sum_{c=k}^{nk-1}\delta_c\frac{c}{d}\Big]\mathbf{x} \\
\Rightarrow \bar{\beta} &= \frac{d}{(1-\delta)nk + \sum_{c=k}^{nk-1}\delta_c c}
\end{aligned} \tag{A.9}
$$

where in (a), $\mathbf{m} \in \mathbb{R}^d$ such that

$$
\mathbf{m}_i = \begin{cases} 1 & \text{if } \Lambda_{jj} > 0 \\ 0 & \text{else.} \end{cases}
$$

Also, by construction of $\boldsymbol{S}$, $\mathrm{rank}(\mathrm{diag}(\mathbf{m})) \leq nk$. Further, (b) follows by symmetry across the d dimensions.

Since $\delta k \leq \sum_{c=k}^{nk-1}\delta_c c \leq \delta(nk-1)$, there is

$$
\frac{d}{(1-\delta)nk + \delta(nk-1)} \leq \bar{\beta} \leq \frac{d}{(1-\delta)nk + \delta k} \tag{A.10}
$$

Computing the MSE. Next, we use the value of $\bar{\beta}$ in Eq. A.9 to compute MSE.

$$
\begin{aligned}
&MSE(\text{Rand-Proj-Spatial(Max)}) = \mathbb{E}[\|\widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial(Max)})} - \bar{\mathbf{x}}\|_2^2] = \mathbb{E}[\|\bar{\beta}\boldsymbol{S}^{\dagger}\boldsymbol{S}\mathbf{x} - \mathbf{x}\|_2^2] \\
&= \bar{\beta}^2\mathbb{E}[\|\boldsymbol{S}^{\dagger}\boldsymbol{S}\mathbf{x}\|_2^2] + \|\mathbf{x}\|_2^2 - 2\Big\langle\bar{\beta}\mathbb{E}[\boldsymbol{S}^{\dagger}\boldsymbol{S}\mathbf{x}], \mathbf{x}\Big\rangle \\
&= \bar{\beta}^2\mathbb{E}[\|\boldsymbol{S}^{\dagger}\boldsymbol{S}\mathbf{x}\|_2^2] - \|\mathbf{x}\|_2^2 \qquad \text{(Using unbiasedness of } \widehat{\mathbf{x}}^{(\text{Rand-Proj-Spatial(Max)})}\text{)} \\
&= \bar{\beta}^2\mathbf{x}^T\mathbb{E}[\boldsymbol{S}^T(\boldsymbol{S}^{\dagger})^T\boldsymbol{S}^{\dagger}\boldsymbol{S}]\mathbf{x} - \|\mathbf{x}\|_2^2.
\end{aligned} \tag{A.11}
$$

Using $\boldsymbol{S}^{\dagger} = \boldsymbol{U}\Lambda^{\dagger}\boldsymbol{U}^T$,

$$
\mathbb{E}[\boldsymbol{S}^T(\boldsymbol{S}^{\dagger})^T\boldsymbol{S}^{\dagger}\boldsymbol{S}] = \mathbb{E}[\boldsymbol{U}\Lambda\boldsymbol{U}^T\boldsymbol{U}\Lambda^{\dagger}\boldsymbol{U}^T\boldsymbol{U}\Lambda^{\dagger}\boldsymbol{U}^T\boldsymbol{U}\Lambda\boldsymbol{U}^T]
$$

$$
\begin{aligned}
&= \mathbb{E}[\boldsymbol{U}\Lambda(\Lambda^\dagger)^2\Lambda\boldsymbol{U}^T] \\
&= \sum_{\boldsymbol{U}=\Phi} \boldsymbol{U}\mathbb{E}[\Lambda(\Lambda^\dagger)^2\Lambda]\boldsymbol{U}^T \cdot \Pr[\boldsymbol{U}=\Phi] \\
&= \sum_{\boldsymbol{U}=\Phi} \boldsymbol{U}\Big[(1-\delta)\frac{nk}{d}\boldsymbol{I}_d + \sum_{c=k}^{nk-1}\delta_c\frac{c}{d}\boldsymbol{I}_d\Big]\boldsymbol{U}^T \cdot \Pr[\boldsymbol{U}=\Phi] \\
&= \Big[(1-\delta)\frac{nk}{d} + \sum_{c=k}^{nk-1}\delta_c\frac{c}{d}\Big] \cdot \sum_{\boldsymbol{U}=\Phi}\boldsymbol{U}\boldsymbol{U}^T \cdot \Pr[\boldsymbol{U}=\Phi] \\
&= \Big[(1-\delta)\frac{nk}{d} + \sum_{c=k}^{nk-1}\delta_c\frac{c}{d}\Big]\boldsymbol{I}_d \\
&= \frac{1}{\bar{\beta}}\boldsymbol{I}_d
\end{aligned} \tag{A.12}
$$

Substituting Eq. A.12 in Eq. A.11, we get

$$
\begin{aligned}
MSE(\text{Rand-Proj-Spatial(Max)}) &= \bar{\beta}^2\mathbf{x}^T\frac{1}{\bar{\beta}}\boldsymbol{I}_d\mathbf{x} - \|\mathbf{x}\|_2^2 = (\bar{\beta}-1)\|\mathbf{x}\|_2^2 \\
&\leq \Big[\frac{d}{(1-\delta)nk+\delta k} - 1\Big]\|\mathbf{x}\|_2^2,
\end{aligned}
$$

where the inequality is by Eq A.10. □

A.3.2 Comparing against Rand-k

Next, we compare the MSE of Rand-Proj-Spatial(Max) with the MSE of the baseline Rand-k analytically in the full-correlation case. Recall that in this case,

$$
MSE(\text{Rand-}k) = \frac{1}{n}(\frac{d}{k}-1)\|\mathbf{x}\|_2^2.
$$

We have

$$
\begin{aligned}
&MSE(\text{Rand-Proj-Spatial(Max)}) \leq MSE(\text{Rand-}k) \\
\Leftrightarrow\ &\frac{d}{(1-\delta)nk+\delta k} - 1 \leq \frac{1}{n}(\frac{d}{k}-1) \\
\Leftrightarrow\ &\frac{d}{k}\frac{n-(1-\delta)n-\delta}{n((1-\delta)n+\delta)} \leq 1 - \frac{1}{n}
\end{aligned}
$$

$$
\begin{aligned}
&\Leftrightarrow \frac{d}{k} \cdot \frac{\delta - \delta/n}{(1-\delta)n + \delta} \leq \frac{n-1}{n} \\
&\Leftrightarrow d\delta(1 - \frac{1}{n})n \leq k(n-1) \cdot ((1-\delta)n + \delta) \\
&\Leftrightarrow d\delta \leq k \cdot ((1-\delta)n + \delta) \\
&\Leftrightarrow d\delta + kn\delta - k\delta \leq kn \\
&\Leftrightarrow \delta \leq \frac{kn}{d + kn - k} \\
&\Leftrightarrow \delta \leq \frac{1}{\frac{d}{kn} + 1 - \frac{1}{n}}
\end{aligned}
$$

Since $nk \leq d$, for $n \geq 2$, the above implies when

$$\delta \leq \frac{1}{1 + \frac{1}{2}} = \frac{2}{3},$$

the MSE of Rand-Proj-Spatial(Max) is always less than that of Rand-k.

A.3.3 $\boldsymbol{S}$ has full rank with high probability

We empirically verify that $\delta \approx 0$. With $d \in \{32, 64, 128, \ldots, 1024\}$ and 4 different nk value such that $nk \leq d$ for each d, we compute rank($\boldsymbol{S}$) for 10^5 trials for each pair of (nk, d) values, and plot the results for all trials. All results are presented in Figure A.2. As one can observe from the plots, rank($\boldsymbol{S}$) $= nk$ with high probability, suggesting $\delta \approx 0$.

This implies the MSE of Rand-Proj-Spatial(Max) is

$$MSE(\text{Rand-Proj-Spatial(Max)}) \approx (\frac{d}{nk} - 1)\|\mathbf{x}\|_2^2,$$

in the full correlation case.

A.3.4 Proof of Theorem 2.4.4

Theorem 4.4 (MSE under No Correlation)**.** *Consider n clients, each holding a vector $\mathbf{x}_i \in \mathbb{R}^d$, $\forall i \in [n]$. Suppose we set $T \equiv 1$, $\bar{\beta} = \frac{d^2}{k}$ in Eq. 2.5, and the random linear*

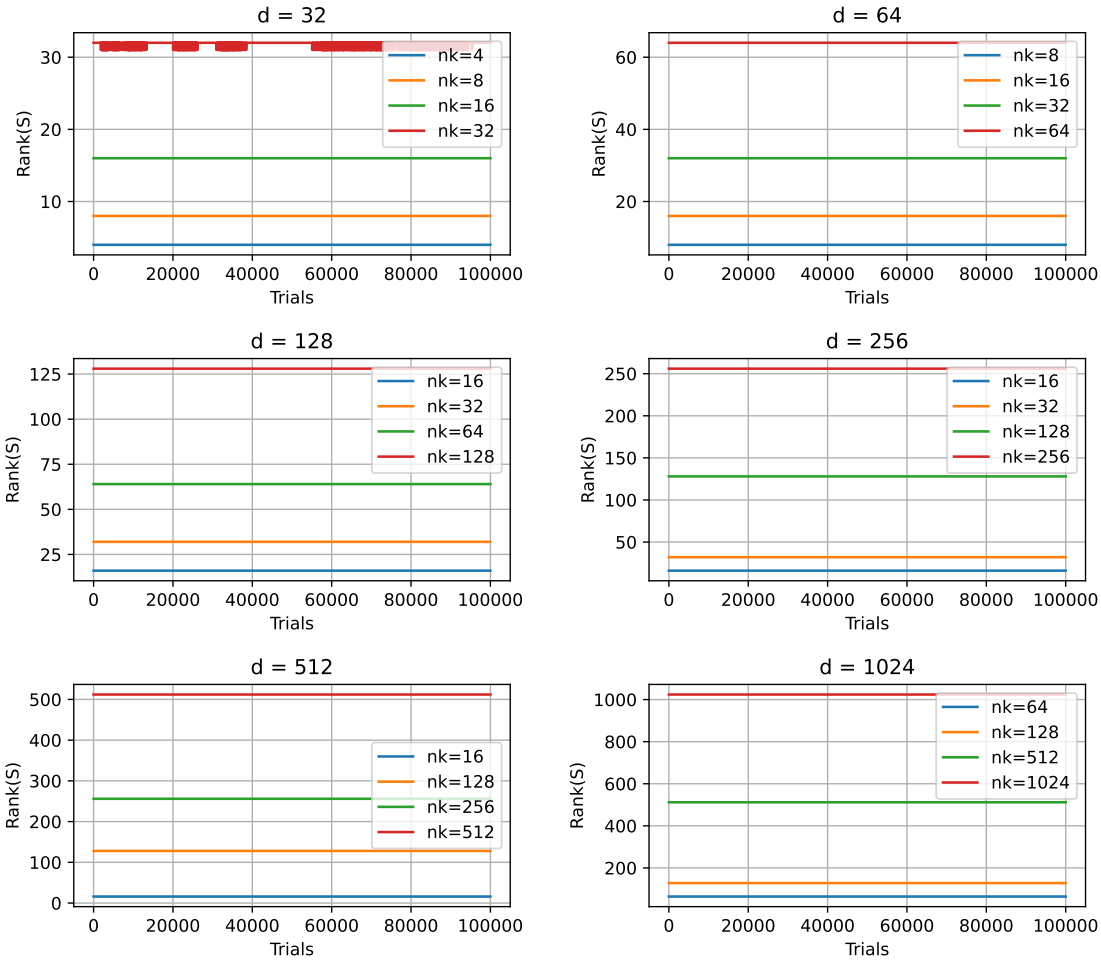

Figure A.2: Simulation results of rank($\boldsymbol{S}$), where $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{G}_i^T \boldsymbol{G}_i$, with $\boldsymbol{G}_i$ being SRHT. With $d \in \{32, 64, 128, \ldots, 1024\}$ and 4 different nk values such that $nk \leq d$ for each d, we compute rank($\mathbf{S}$) for 10^5 trials for each pairs of (nk, d) values and plot the results for all trials. When $d = 32$ and $nk = 32$ in the first plot, rank($\boldsymbol{S}$) $= 31$ in 2100 trials, and rank($\boldsymbol{S}$) $= nk = 32$ in all the rest of the trials. For all other (nk, d) pairs, $\boldsymbol{S}$ always has rank nk in the 10^5 trials. This verifies that $\delta = \Pr[\text{rank}(\boldsymbol{S}) < nk] \approx 0$.

map $\boldsymbol{G}_i$ *at each client to be an SRHT matrix. Then, for* $nk \leq d$*,*

$$\mathbb{E}\Big[\|\widehat{\mathbf{x}}^{(\textit{Rand-Proj-Spatial})} - \bar{\mathbf{x}}\|_2^2\Big] = \frac{1}{n^2}\Big(\frac{d}{k} - 1\Big)\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2.$$

Proof. When the client vectors are all orthogonal to each other, we define the transformation function on the eigenvalue to be $T(\lambda) = 1, \forall\lambda \geq 0$. We show that by considering the above constant T, SRHT becomes the same as rand k. Recall $\boldsymbol{S} = \sum_{i=1}^{n}\boldsymbol{G}_i^T\boldsymbol{G}_i$ and let $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$ be its eigendecompostion. Then,

$$T(\boldsymbol{S}) = \boldsymbol{U}T(\Lambda)\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{I}_d\boldsymbol{U}^T = \boldsymbol{I}_d.$$

Hence, $(T(\boldsymbol{S}))^\dagger = \boldsymbol{I}_d$. And the decoded vector for client i becomes

$$\begin{aligned}
\widehat{\mathbf{x}}_i &= \bar{\beta}\Big(T(\boldsymbol{G}^T\boldsymbol{G})\Big)^\dagger\boldsymbol{G}_i^T\boldsymbol{G}_i\mathbf{x}_i = \bar{\beta}\boldsymbol{G}_i^T\boldsymbol{G}_i\mathbf{x}_i = \bar{\beta}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i, \\
\widehat{\mathbf{x}} &= \frac{1}{n}\sum_{i=1}^{n}\widehat{\mathbf{x}}_i = \frac{1}{n}\bar{\beta}\sum_{i=1}^{n}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i
\end{aligned} \tag{A.13}$$

$\boldsymbol{D}_i$ is a diagonal matrix. Also, $\boldsymbol{E}_i^T\boldsymbol{E}_i \in \mathbb{R}^{d\times d}$ is a diagonal matrix, where the i-th entry is 0 or 1.

Computing $\bar{\beta}$**.** To ensure that $\widehat{\mathbf{x}}$ is an unbiased estimator, from Eq. A.13

$$\begin{aligned}
\mathbf{x}_i &= \bar{\beta}\mathbb{E}[\boldsymbol{G}_i^T\boldsymbol{G}_i]\mathbf{x}_i \\
&= \frac{\bar{\beta}}{d}\mathbb{E}[\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i]\mathbf{x}_i \\
&= \frac{\bar{\beta}}{d}\mathbb{E}_{\boldsymbol{D}_i}\Big[\boldsymbol{D}_i\boldsymbol{H}^T\underbrace{\mathbb{E}[\boldsymbol{E}_i^T\boldsymbol{E}_i]}_{=(k/d)\boldsymbol{I}_d}\boldsymbol{H}\boldsymbol{D}_i\Big]\mathbf{x}_i && (\because \boldsymbol{E}_i \text{ is independent of } \boldsymbol{D}_i) \\
&= \frac{\bar{\beta}}{d}k\mathbb{E}_{\boldsymbol{D}_i}\left[\boldsymbol{D}_i^2\right]\mathbf{x}_i && (\because \boldsymbol{H}^T\boldsymbol{H} = d\boldsymbol{I}_d) \\
&= \frac{\bar{\beta}k}{d}\mathbf{x}_i && (\because \boldsymbol{D}_i^2 = \boldsymbol{I} \text{ is now deterministic.}) \\
\Rightarrow \bar{\beta} &= \frac{d}{k}.
\end{aligned} \tag{A.14}$$

Computing the MSE.

$$
\begin{aligned}
MSE &= \mathbb{E}\Big\|\widehat{\mathbf{x}} - \bar{\mathbf{x}}\Big\|_2^2 \\
&= \mathbb{E}\Big\|\frac{1}{n}\bar{\beta}\sum_{i=1}^{n}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\Big\|_2^2 \\
&= \frac{1}{n^2}\Bigg\{\mathbb{E}\Big\|\bar{\beta}\sum_{i=1}^{n}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i\Big\|_2^2 + \Big\|\sum_{i=1}^{n}\mathbf{x}_i\Big\|_2^2 \\
&\qquad\qquad -2\Big\langle\bar{\beta}\mathbb{E}[\sum_{i=1}^{n}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i], \sum_{i=1}^{n}\mathbf{x}_i\Big\rangle\Bigg\} \\
&= \frac{1}{n^2}\Bigg\{\bar{\beta}^2\mathbb{E}\Big\|\sum_{i=1}^{n}\frac{1}{d}\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i\Big\|_2^2 - \Big\|\sum_{i=1}^{n}\mathbf{x}_i\Big\|_2^2\Bigg\} \qquad (\because \mathbb{E}[\widehat{\mathbf{x}}] = \bar{\mathbf{x}}) \\
&= \frac{1}{n^2}\Bigg\{\sum_{i=1}^{n}\frac{\bar{\beta}^2}{d^2}\mathbb{E}\Big\|\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i\Big\|_2^2 - \sum_{i=1}^{n}\Big\|\mathbf{x}_i\Big\|_2^2 \qquad \text{(A.15)} \\
&\quad +2\sum_{i=1}^{n}\sum_{l=i+1}^{n}\frac{\bar{\beta}^2}{d^2}\Big\langle\mathbb{E}[\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i], \mathbb{E}[\boldsymbol{D}_l\boldsymbol{H}^T\boldsymbol{E}_l^T\boldsymbol{E}_l\boldsymbol{H}\boldsymbol{D}_l\mathbf{x}_l]\Big\rangle - 2\sum_{i=1}^{n}\sum_{l=i+1}^{n}\Big\langle\mathbf{x}_i, \mathbf{x}_l\Big\rangle\Bigg\}.
\end{aligned}
$$

Note that in Eq. A.15

$$
\begin{aligned}
&\mathbb{E}\Big\|\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i\Big\|_2^2 = \mathbb{E}[\mathbf{x}_i^T\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i] \\
&= d\mathbb{E}[\mathbf{x}_i^T\boldsymbol{D}_i\boldsymbol{H}^T(\boldsymbol{E}_i^T\boldsymbol{E}_i)^2\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i] \qquad (\because \boldsymbol{D}_i^2 = \boldsymbol{I}_d; \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{H}\boldsymbol{H}^T = d\boldsymbol{I}_d) \\
&= d\mathbf{x}_i^T\mathbb{E}_{\boldsymbol{D}_i}\left[\boldsymbol{D}_i\boldsymbol{H}^T\mathbb{E}[\boldsymbol{E}_i^T\boldsymbol{E}_i]\boldsymbol{H}\boldsymbol{D}_i\right]\mathbf{x}_i \quad (\boldsymbol{E}_i, \boldsymbol{D}_i \text{ are independent; } (\boldsymbol{E}_i^T\boldsymbol{E}_i)^2 = \boldsymbol{E}_i^T\boldsymbol{E}_i) \\
&= kd\|\mathbf{x}_i\|_2^2, \qquad \text{(A.16)}
\end{aligned}
$$

since $\mathbb{E}[\boldsymbol{E}_i^T\boldsymbol{E}_i] = (k/d)\boldsymbol{I}_d$, $\boldsymbol{H}^T\boldsymbol{H} = d\boldsymbol{I}_d$ and for $i \neq l$

$$
\Big\langle\mathbb{E}[\boldsymbol{D}_i\boldsymbol{H}^T\boldsymbol{E}_i^T\boldsymbol{E}_i\boldsymbol{H}\boldsymbol{D}_i\mathbf{x}_i], \mathbb{E}[\boldsymbol{D}_l\boldsymbol{H}^T\boldsymbol{E}_l^T\boldsymbol{E}_l\boldsymbol{H}\boldsymbol{D}_l\mathbf{x}_l]\Big\rangle = \Big\langle k\mathbf{x}_i, k\mathbf{x}_l\Big\rangle = k^2\Big\langle\mathbf{x}_i, \mathbf{x}_l\Big\rangle. \qquad \text{(A.17)}
$$

Substituting Eq. A.16, A.17 in Eq. A.15, we get

$$
MSE = \frac{1}{n^2}\Bigg\{\Big(\frac{\bar{\beta}^2}{d^2}\sum_{i=1}^{n}kd\|\mathbf{x}_i\|_2^2 + 2\sum_{i=1}^{n}\sum_{l=i+1}^{n}\frac{\bar{\beta}^2k^2}{d^2}\Big\langle\mathbf{x}_i, \mathbf{x}_l\Big\rangle\Big) - \sum_{i=1}^{n}\Big\|\mathbf{x}_i\Big\|_2^2 - 2\sum_{i=1}^{n}\sum_{l=i+1}^{n}\Big\langle\mathbf{x}_i, \mathbf{x}_l\Big\rangle\Bigg\}
$$

$$= \frac{1}{n^2}\left(\frac{d}{k} - 1\right)\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2,$$

which is exactly the same as the MSE of rand k. □

A.3.5 Rand-Proj-Spatial recovers Rand-k-Spatial (Proof of Lemma 4.1)

Lemma 4.1 (Recovering Rand-k-Spatial)**.** *Suppose client i generates a subsampling matrix $\boldsymbol{E}_i = \left[\mathbf{e}_{i_1}, \ \ldots, \ \mathbf{e}_{i_k}\right]^\top$, where $\{\mathbf{e}_j\}_{j=1}^d$ are the canonical basis vectors, and $\{i_1, \ldots, i_k\}$ are sampled from $\{1, \ldots, d\}$ without replacement. The encoded vectors are given as $\widehat{\mathbf{x}}_i = \boldsymbol{E}_i\mathbf{x}_i$. Given a function T, $\widehat{\mathbf{x}}$ computed as in Eq. 2.5 recovers the Rand-k-Spatial estimator.*

Proof. If client i applies $\boldsymbol{E}_i \in \mathbb{R}^{k\times d}$ as the random matrix to encode $\mathbf{x}_i$ in Rand-Proj-Spatial, by Eq. 2.5, client i's encoded vector is now

$$\hat{\mathbf{x}}_i^{(\text{Rand-Proj-Spatial})} = \bar{\beta}\left(T(\sum_{i=1}^{n} \boldsymbol{E}_i^T\boldsymbol{E}_i)\right)^\dagger \boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i \tag{A.18}$$

Notice $\boldsymbol{E}_i^T\boldsymbol{E}_i$ is a diagonal matrix, where the j-th diagonal entry is 1 if coordinate j of $\mathbf{x}_i$ is chosen. Hence, $\boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i$ can be viewed as choosing k coordinates of $\mathbf{x}_i$ without replacement, which is exactly the same as Rand-k-Spatial's (and Rand-k's) encoding procedure.

Notice $\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i$ is also a diagonal matrix, where the j-th diagonal entry is exactly M_j, i.e. the number of clients who selects the j-th coordinate as in Rand-k-Spatial [52]. Furthermore, notice $\left(T(\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i)\right)^\dagger$ is also a diagonal matrix, where the j-th diagonal entry is $\frac{1}{T(M_j)}$, which recovers the scaling factor used in Rand-k-Spatial's decoding procedure.

Rand-Proj-Spatial computes $\bar{\beta}$ as $\bar{\beta}\mathbb{E}\left[\left(T(\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i)\right)^\dagger \boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i\right] = \mathbf{x}_i$. Since $\left(T(\sum_{i=1}^n \boldsymbol{E}_i^T\boldsymbol{E}_i)\right)^\dagger$ and $\boldsymbol{E}_i^T\boldsymbol{E}_i\mathbf{x}_i$ recover the scaling factor and the encoding procedure of Rand-k-Spatial, and $\bar{\beta}$ is computed in exactly the same way as Rand-k-Spatial does, $\bar{\beta}$ will be exactly the same as in Rand-k-Spatial.

Therefore, $\hat{\mathbf{x}}_i^{(\text{Rand-Proj-Spatial})}$ in Eq. A.18 with $\boldsymbol{E}_i$ as the random matrix at client i recovers $\hat{\mathbf{x}}_i^{(\text{Rand-}k\text{-Spatial})}$. This implies Rand-Proj-Spatial recovers Rand-k-Spatial in this case. □

A.4 Additional Experiment Details and Results

Implementation. All experiments are conducted in a cluster of 20 machines, each of which has 40 cores. The implementation is in `Python`, mainly based on `numpy` and `scipy`. All code used for the experiments can be found at https://github.com/11hifish/Rand-Proj-Spatial.

Data Split. For the non-IID dataset split across the clients, we follow [73] to split `Fashion-MNIST`, which is used in distributed power iteration and distributed k-means. Specifically, the data is first sorted by labels and then divided into $2n$ shards with each shard corresponding to the data of a particular label. Each client is then assigned 2 shards (i.e., data from 2 classes). However, this approach only works for datasets with discrete labels (i.e. datasets used in classification tasks). For the other dataset `UJIndoor`, which is used in distributed linear regression, we first sort the dataset by the ground truth prediction and then divides the sorted dataset across the clients.

A.4.1 Additional experimental results

For each one of the three tasks, distributed power iteration, distributed k-means, and distributed linear regression, we provide additional results when the data split is IID across the clients for smaller n, k values in Section A.4.1, and when the data split is Non-IID across the clients in Section A.4.1. For the Non-IID case, we use the same settings (i.e. n, k, d values) as in the IID case.

Discussion. For smaller n, k values compared to the data dimension d, there is less information or less correlation from the client vectors. Hence, both Rand-k-Spatial and Rand-Proj-Spatial perform better as nk increases. When n, k is small, one might notice Rand-Proj-Spatial performs worse than Rand-k-Wangni in some settings. However, Rand-k-Wangni is an *adaptive* estimator, which optimizes the sampling weights for choosing the client vector coordinates through an iterative process. That means Rand-k-Wangni requires more computation from the clients, while in practice, the clients often have limited computational power. In contrast, our Rand-Proj-Spatial estimator is *non-adaptive* and the server does more computation instead of the clients. This is more practical since the central server usually has more computational power

than the clients in applications like FL. See the introduction for more discussion.

In most settings, we observe the proposed Rand-Proj-Spatial has a better performance compared to Rand-k-Spatial. Furthermore, as one would expect, both Rand-k-Spatial and Rand-Proj-Spatial perform better when the data split is IID across the clients since there is more correlation among the client vectors in the IID case than in the Non-IID case.

More results in the IID case

Distributed Power Iteration and Distribued K-Means. We use the `Fashion-MNIST` dataset for both distributed power iteration and distributed k-means, which has a dimension of $d = 1024$. We consider more settings for distributed power iteration and distributed k-means here: $n = 10, k \in \{5, 25, 51\}$, and $n = 50, k \in \{5, 10\}$.

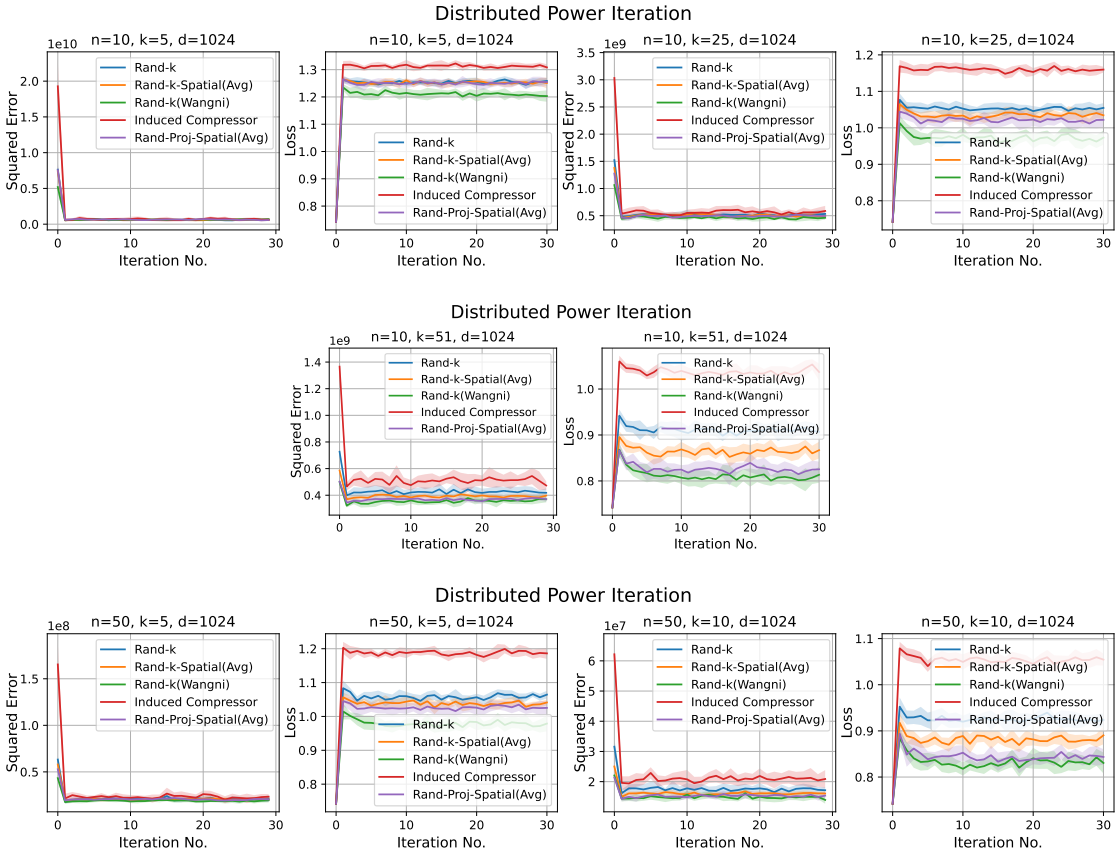

Figure A.3: More results of distributed power iteration on `Fashion-MNIST` (IID data split) with $d = 1024$ when $n = 10$, $k \in \{5, 25, 51\}$ and when $n = 50$, $k \in \{5, 10\}$.

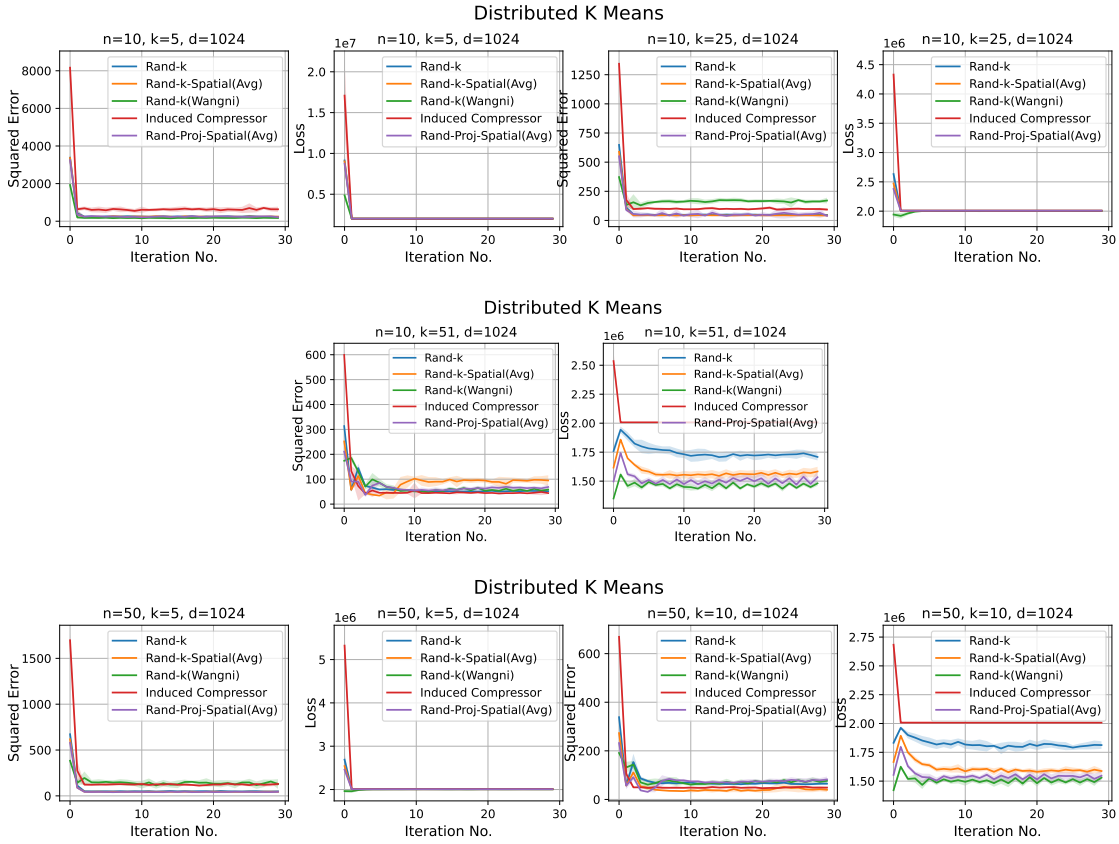

Figure A.4: More results on distributed k-means on `Fashion-MNIST` (IID data split) with $d = 1024$ when $n = 10, k \in \{5, 25, 51\}$ and when $n = 50, k \in \{10, 51\}$.

Distributed Linear Regression. We use the `UJIndoor` dataset distributed linear regression, which has a dimension of $d = 512$. We consider more settings here: $n = 10, k \in \{5, 25\}$ and $n = 50, k \in \{1, 5\}$.

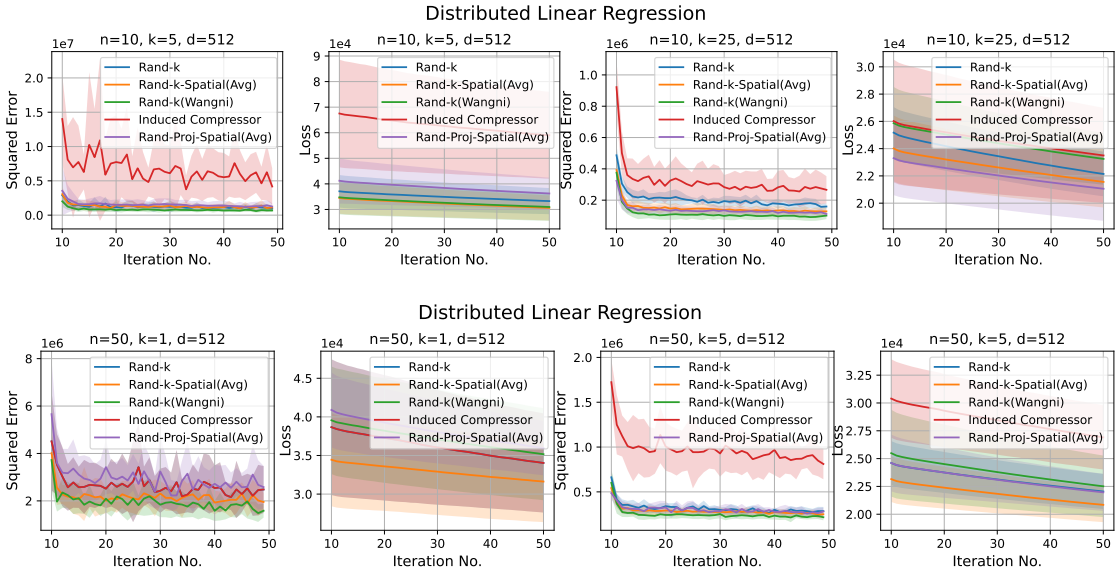

Figure A.5: More results of distributed linear regression on `UJIndoor` (IID data split) with $d = 512$, when $n = 10, k \in \{5, 25\}$ and when $n = 50, k \in \{1, 5\}$. Note when $k = 1$, the Induced estimator is the same as Rand-k.

Additional results in the Non-IID case

In this section, we report results when the dataset split across the clients are Non-IID, using the same datasets as in the IID case. We choose exactly the same set of n, k values as in the IID case.

Distributed Power Iteration and Distributed K-Means. Again, both distributed power iteration and distributed k-means use the `Fashion-MNIST` dataset, with a dimension $d = 1024$. We consider the following settings for both tasks: $n = 10, k \in \{5, 25, 51, 102\}$ and $n = 50, k \in \{5, 10, 20\}$.

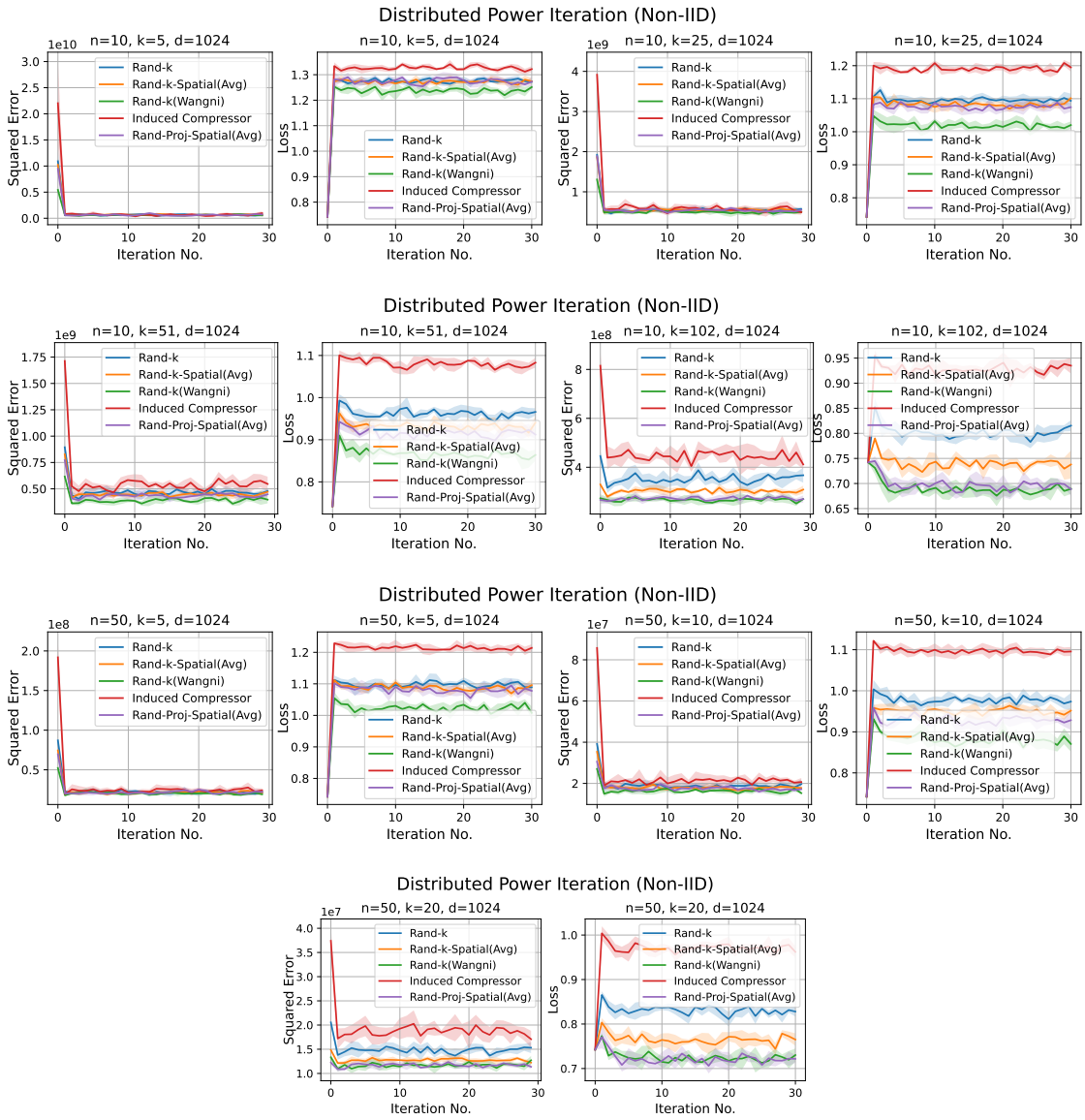

Figure A.6: Results of distributed power iteration when the data split is Non-IID. $n = 10, k \in \{5, 25, 51, 102\}$ and $n = 50, k \in \{5, 10, 20\}$.

Distributed Linear Regression. Again, we use the `UJIndoor` dataset for distributed linear regression, which has a dimension $d = 512$. We consider the following settings: $n = 10, k \in \{5, 25, 50\}$ and $n = 50, k \in \{1, 5, 50\}$.

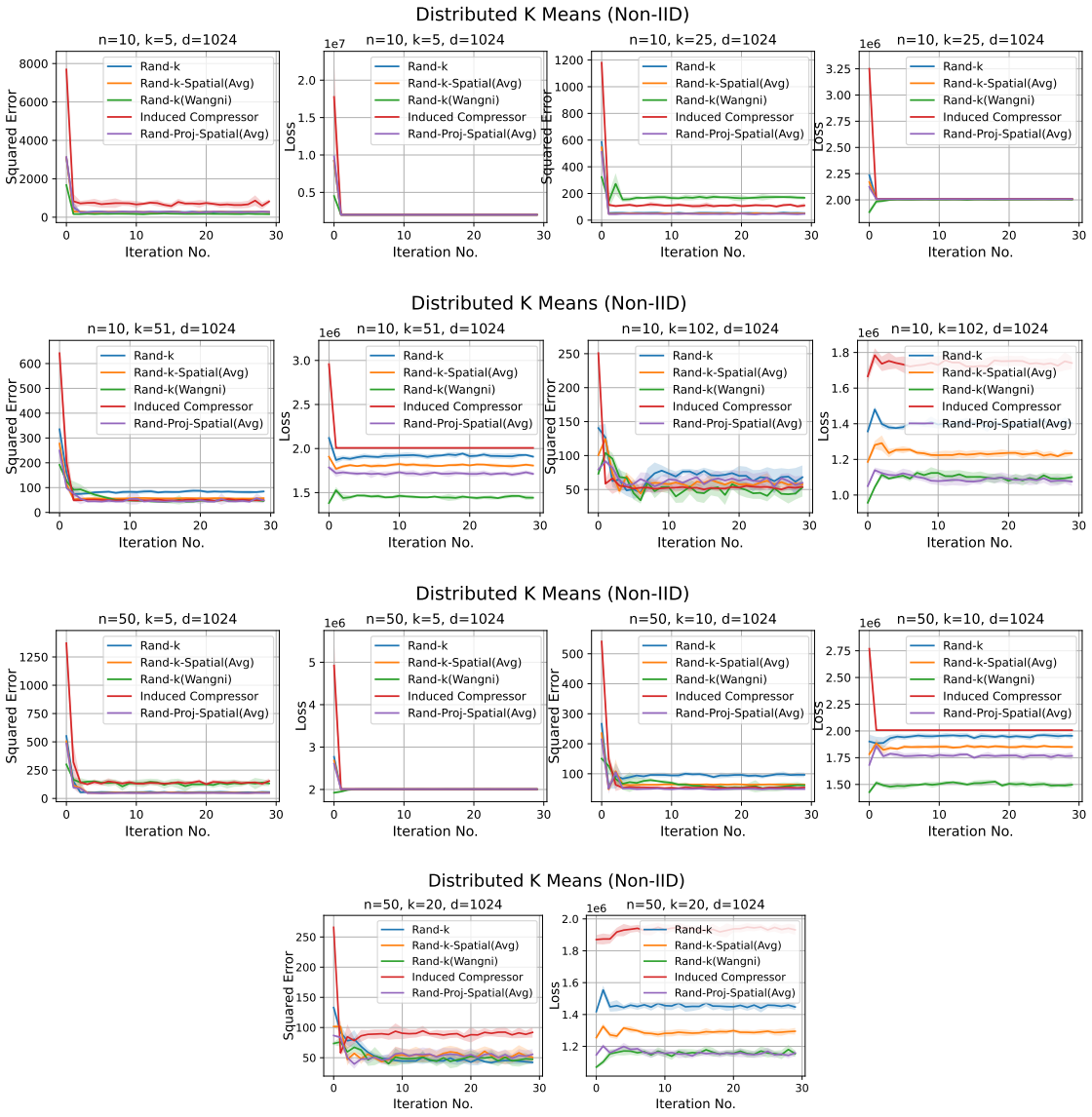

Figure A.7: Results of distributed k-means when the data split is Non-IID. $n = 10, k \in \{5, 25, 51, 102\}$ and $n = 50, k \in \{5, 10, 20\}$.

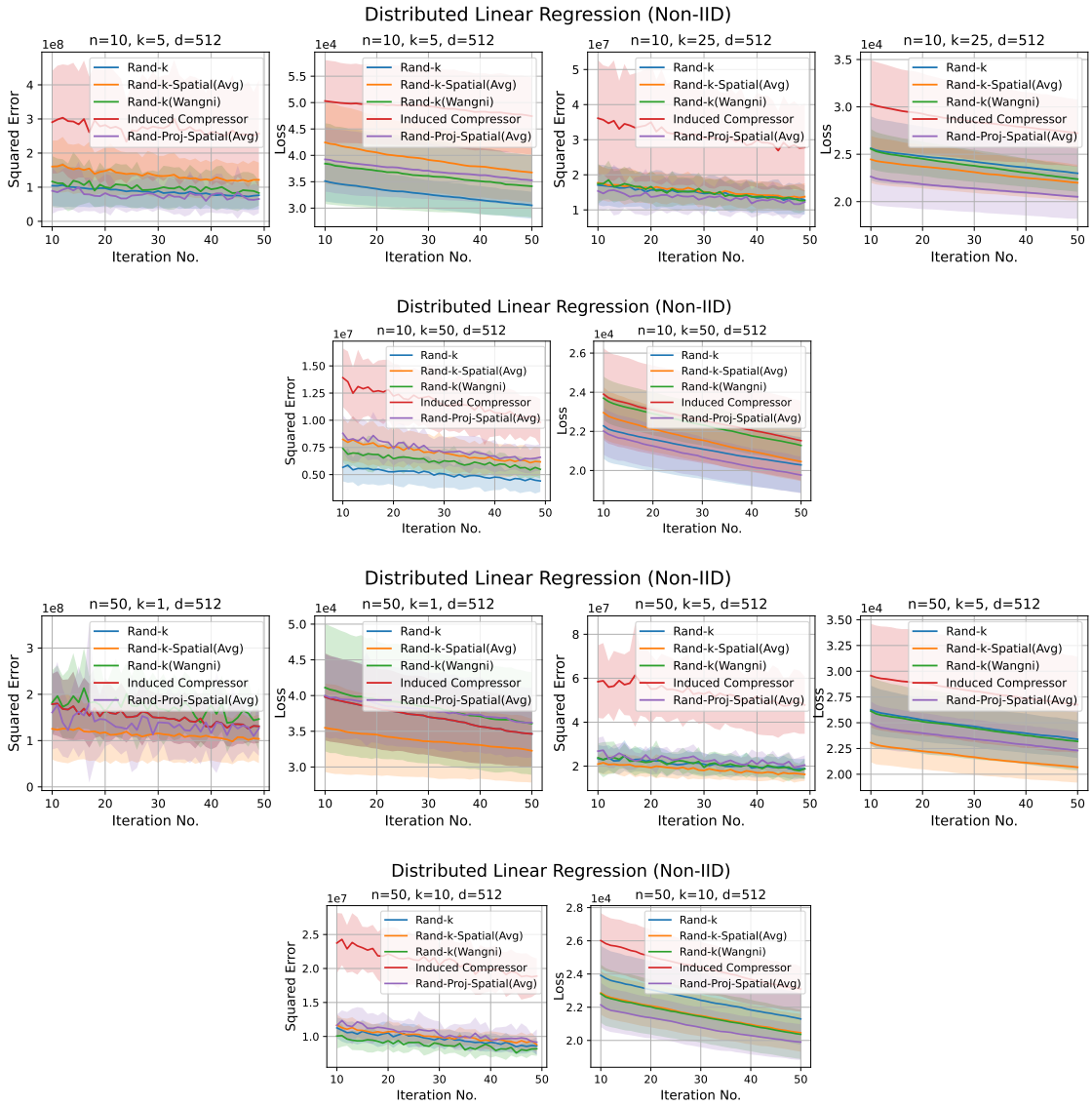

Figure A.8: Results of distributed linear regression when the data split is Non-IID. $n = 10, k \in \{5, 25, 50\}$ and $n = 50, k \in \{1, 5, 50\}$.

Appendix B

Private Majority Ensembling

B.1 Details of Section 3.3

B.1.1 Randomized Response with Constant Probability p_{const}

We show the magnitude of p_{const} in RR (Algorithm 3) to solve Problem 3.1.1, such that the output is $(m\epsilon, \delta)$-DP, in Lemma B.1.1.

Lemma B.1.1. *Consider using RR (Algorithm 3) to solve Problem 3.1.1. Let the majority of K (ϵ, Δ)-differentially private mechanisms be $(\tau\epsilon, \lambda)$-differentially private, where $\tau \in [1, K]$ and $\lambda \in [0, 1)$ are computed by simple composition (Theorem 3.2.2) or general composition (Theorem 3.2.3). If*

$$p_{const} \leq \frac{e^{m\epsilon} - 1 + 2\delta}{\frac{2(e^{\tau\epsilon} - e^{m\epsilon} + (1+e^{m\epsilon})\lambda)}{e^{\tau\epsilon}+1} + e^{m\epsilon} - 1} \tag{B.1}$$

then RR is $(m\epsilon, \delta)$-differentially private.

Proof of Lemma B.1.1. Let $x \in \{0, 1\}$ denote the output of RR. Let $q_x = \Pr[\mathcal{L}(\mathcal{D}) = x]$ and $q'_x = \Pr[\mathcal{L}(\mathcal{D}') = x]$, where $\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{K} M_i(\mathcal{D})$, $\mathcal{L}(\mathcal{D}') = \sum_{i=1}^{K} M_i(\mathcal{D}')$ and $\mathcal{D}$, $\mathcal{D}'$ are adjacent datasets. Recall each mechanism M_i is (ϵ, Δ)-differentially private, and the majority of the outputs of $\{M_i\}_{i=1}^{K}$ is $(\tau\epsilon, \lambda)$-differentially private. When $\Delta = 0$, using simple composition, $\tau = K$ and $\lambda = 0$. When $\Delta > 0$, using general composition $\tau \approx \sqrt{K}$ and $\lambda \approx K\Delta$. By definition of differential privacy

Algorithm 3 Randomized Response Majority (RR)

Input: K (ϵ, Δ)-DP mechanisms $\{M_i\}_{i=1}^K$, noise function $\gamma : \{0, \dots, K\} \to [0, 1]$, dataset $\mathcal{D}$, privacy allowance $1 \leq m \leq K$, failure probability $\delta \geq \Delta \geq 0$
Output: $(m\epsilon, \delta)$-DP majority vote of $\{M_i\}_{i=1}^K$
Compute a *constant* probability $p_{const} \in [0, 1]$
Flip the p_{const}- biased coin
if Head (with probability p_{const}) **then**
    $\mathcal{S} = \{S_1, .., S_k\}$, where $S_i \sim M_i(\mathcal{D})$
    $\mathcal{L} = \sum_{i=1}^K S_i$
    Output $\mathbb{I}\{\frac{1}{K}\mathcal{L} \geq \frac{1}{2}\}$
else
    Output 0/1 with equal probability
end if

(Definition 4.2.1), all of the following four constraints on q_x, q'_x apply:

$$q_x \leq e^{\tau\epsilon} q'_x + \lambda, \quad \text{and} \quad 1 - q'_x \leq e^{\tau\epsilon}(1 - q_x) + \lambda$$
$$q'_x \leq e^{\tau\epsilon} q_x + \lambda, \quad \text{and} \quad 1 - q_x \leq e^{\tau\epsilon}(1 - q'_x) + \lambda$$

To ensure RR is $(m\epsilon, \delta)$-differentially private, p_{const} needs to be such that for all possible $q_x, q'_x \in [0, 1]$,

$$\Pr[\mathsf{RR}(\mathcal{D}) = x] \leq e^{m\epsilon} \Pr[\mathsf{RR}(\mathcal{D}') = x] + \delta \tag{B.2}$$
$$p_{const} \cdot q_x + \frac{1}{2}(1 - p_{const}) \leq e^{m\epsilon}(p_{const} \cdot q'_x + \frac{1}{2}(1 - p_{const})) + \delta \tag{B.3}$$
$$(q_x - e^{m\epsilon} q'_x + \frac{1}{2} e^{m\epsilon} - \frac{1}{2}) \cdot p_{const} \leq \frac{1}{2} e^{m\epsilon} - \frac{1}{2} + \delta \tag{B.4}$$

Let $h(q_x, q'_x) := q_x - e^{m\epsilon} q'_x + \frac{1}{2} e^{m\epsilon} - \frac{1}{2}$. The above inequality of p_{const} (Eq. B.4) needs to hold for worst case output probabilities q^*_x, q'^*_x that cause the maximum privacy loss. That is, p_{const} needs to satisfy

$$p_{const} \cdot max_{q_x, q'_x} h(q_x, q'_x) \leq \frac{1}{2} e^{m\epsilon} - \frac{1}{2} + \delta \tag{B.5}$$

To find the worst case output probabilities, we solve the following Linear Pro-

gramming (LP) problem:

$$\text{Objective:} \quad \max_{q_x, q'_x} \quad h(q_x, q'_x) := q_x - e^{m\epsilon} q'_x + \frac{1}{2} e^{m\epsilon} - \frac{1}{2} \tag{B.6}$$

$$\text{Subject to:} \quad 0 \leq q_x \leq 1, 0 \leq q'_x \leq 1 \tag{B.7}$$

$$q_x \leq e^{\tau\epsilon} q'_x + \lambda, 1 - q'_x \leq e^{\tau\epsilon}(1 - q_x) + \lambda \tag{B.8}$$

$$q'_x \leq e^{\tau\epsilon} q_x + \lambda, 1 - q_x \leq e^{\tau\epsilon}(1 - q'_x) + \lambda \tag{B.9}$$

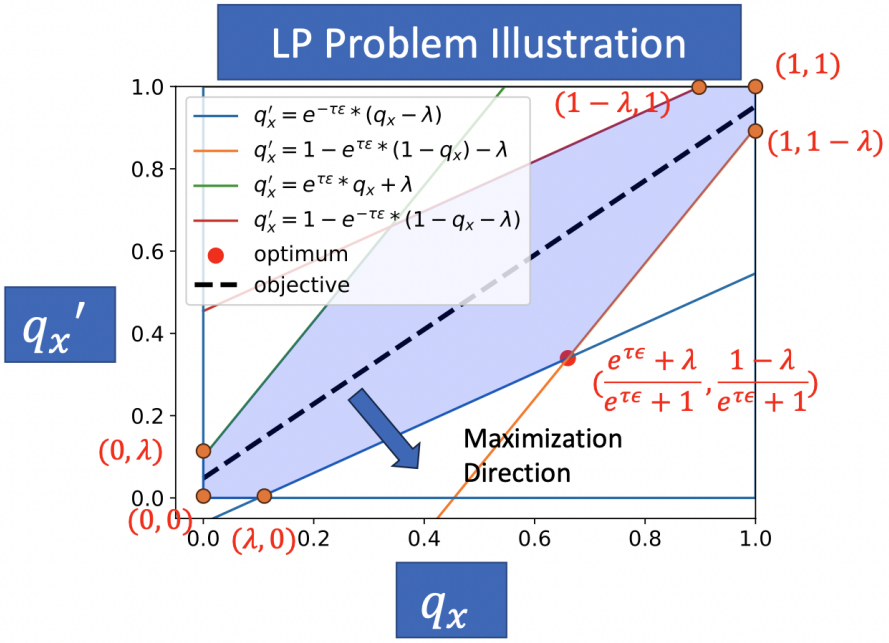

Figure B.1: A visualization of the above LP problem.

The optimum of any LP problem is at the corners of the feasible region, which is bounded by the optimization constraints. We plot the feasible region $\mathcal{F}$ and the objective of the above LP problem in Figure B.1. Here, $(q_x^*, q_x'^*) = \text{argmax}_{q_x, q'_x} h(q_x, q'_x) \in \{(0,0), (1,1), (0,\lambda), (\lambda,0), (1-\lambda,1), (1,1-\lambda), (\frac{1-\lambda}{e^{\tau\epsilon}+1}, \frac{e^{\tau\epsilon}+\lambda}{e^{\tau\epsilon}+1}), (\frac{e^{\tau\epsilon}+\lambda}{e^{\tau\epsilon}+1}, \frac{1-\lambda}{e^{\tau\epsilon}+1})\}$. The optimum of the LP problem – that is, the worse case probabilities $q_x^*, q_x'^*$ – is,

$$q_x^* = \frac{e^{\tau\epsilon} + \lambda}{e^{\tau\epsilon} + 1}, \quad q_x'^* = \frac{1 - \lambda}{e^{\tau\epsilon} + 1} \tag{B.10}$$

Algorithm 4 Subsampling Majority (SubMaj)

1: Input: K (ϵ, Δ)-DP mechanisms $\{M_i\}_{i=1}^K$, noise function $\gamma : \{0, \dots, K\} \to [0, 1]$, dataset $\mathcal{D}$, privacy allowance $1 \le m \le K$, failure probability $\delta \ge \Delta \ge 0$
2: Output: $(m\epsilon, \delta)$-DP majority vote of $\{M_i\}_{i=1}^K$
3: $\mathcal{S} = \{S_1, .., S_k\}$, where $S_i \sim M_i(\mathcal{D})$
4: $\mathcal{J}_m \leftarrow m$ indices chosen uniformly at random from $[K]$ without replacement
5: $\widehat{\mathcal{L}} = \sum_{j \in \mathcal{J}} S_j$
6: Output $\mathbb{I}\{\frac{1}{m}\widehat{\mathcal{L}} \ge \frac{1}{2}\}$

By Eq. B.5,

$$p_{const} \cdot \left(\frac{e^{\tau\epsilon} + \lambda}{e^{\tau\epsilon} + 1} - e^{m\epsilon} \frac{1 - \lambda}{e^{\tau\epsilon} + 1} + \frac{1}{2} e^{m\epsilon} - \frac{1}{2} \right) \le \frac{1}{2}(e^{m\epsilon} - 1) + \delta \tag{B.11}$$

$$p_{const} \cdot \left(\frac{e^{\tau\epsilon} - e^{m\epsilon} + (1 + e^{m\epsilon})\lambda}{e^{\tau\epsilon} + 1} + \frac{1}{2}(e^{m\epsilon} - 1) \right) \le \frac{1}{2}(e^{m\epsilon} - 1) + \delta \tag{B.12}$$

$$p_{const} \le \frac{e^{m\epsilon} - 1 + 2\delta}{\frac{2(e^{\tau\epsilon} - e^{m\epsilon} + (1 + e^{m\epsilon})\lambda)}{e^{\tau\epsilon} + 1}} + e^{m\epsilon} - 1 \tag{B.13}$$

For small m, ϵ, K, using the approximation $e^y \approx 1 + y$ and that $\tau\epsilon < 2$,

$$p_{const} \approx \frac{m\epsilon + 2\delta}{\frac{2(\tau\epsilon - m\epsilon + (2 + m\epsilon)\lambda)}{\tau\epsilon + 2}} + m\epsilon \approx \frac{m\epsilon + 2\delta}{\tau\epsilon + (2 + m\epsilon)\lambda} \tag{B.14}$$

In the pure differential privacy setting, $\delta = 0, \lambda = 0, \tau = K$, and so $p_{const} \approx \frac{m}{K}$; and in the approximate differential privacy setting, $\lambda \approx 0, \delta \approx 0, \tau \approx \sqrt{K}$, and so $p_{const} \approx \frac{m}{\sqrt{K}}$. □

B.1.2 Proof of Lemma 3.3.1

Lemma B.1.2 (Restatement of Lemma 3.3.1)**.** *Consider Problem 3.1.1, with the privacy allowance $m \in [K]$. Consider the data-dependent algorithm that computes $\mathcal{L}(\mathcal{D})$ and then applies RR with probability p_γ. If $p_\gamma = \gamma_{Sub}(l)$, where $l \in \{0, 1, \dots, K\}$ is the value of $\mathcal{L}(\mathcal{D})$, i.e., the (random) sum of observed outcomes on dataset $\mathcal{D}$, and*

$\gamma_{Sub} : \{0, 1, \ldots, K\} \to [0, 1]$ *is*

$$\gamma_{Sub}(l) = \gamma_{Sub}(K - l)$$
$$= \begin{cases} 1 - 2\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} & \text{if } m \text{ is odd} \\ 1 - 2\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} & \text{if } m \text{ is even} \end{cases}$$

then the majority of m out of K subsampled mechanisms without replacement and the output of our data-dependent RR algorithm have the same distribution.

Proof of Lemma 3.3.1. Let $\mathcal{L} = \sum_{i=1}^{K} S_i$ be the sum of observed outcomes from K mechanisms. Following Algorithm 4, $\mathcal{J}_m$ denotes the m indices chosen uniformly at random from $[K]$ without replacement. Conditioned on $\mathcal{L}$, notice the output of SubMaj follows a hypergeometric distribution. The output probability of SubMaj is

$$\Pr[\mathsf{SubMaj}(\mathcal{D}) = 1] = \sum_{l=0}^{K} \Pr[\mathsf{SubMaj}(\mathcal{D}) = 1 \mid \mathcal{L} = l] \cdot \Pr[\mathcal{L} = l] \tag{B.15}$$
$$= \sum_{l=0}^{K} \Pr[\sum_{j \in \mathcal{J}_m} S_j \geq \frac{m}{2} \mid \mathcal{L} = l] \cdot \Pr[\mathcal{L} = l] \tag{B.16}$$
$$= \begin{cases} \sum_{l=0}^{K} (\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}}) \cdot \Pr[\mathcal{L} = l] & \text{if } m \text{ is odd} \\ \sum_{l=0}^{K} (\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} + \frac{1}{2}\frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}}) \cdot \Pr[\mathcal{L} = l] & \text{if } m \text{ is even} \end{cases} \tag{B.17}$$

Consider an arbitrary noise function $\gamma_{Sub} : \{0, 1, \ldots, K\} \to [0, 1]$. Let $\mathsf{RR\text{-}d}(\mathcal{D})$ denote the output of the data-dependent RR-d on dataset $\mathcal{D}$, where RR-d has the *non-constant* probability set by γ_{Sub}. The output probability of RR is,

$$\Pr[\mathsf{RR\text{-}d}(\mathcal{D}) = 1] = \sum_{l=0}^{K} \Pr[\mathsf{RR\text{-}d}(\mathcal{D}) = 1 \mid \mathcal{L} = l] \cdot \Pr[\mathcal{L} = l] \tag{B.18}$$
$$= \sum_{l=0}^{K} (\gamma_{Sub}(l) \cdot \mathbb{I}\{l \geq \frac{K+1}{2}\} + \frac{1}{2}(1 - \gamma_{Sub}(l))) \cdot \Pr[\mathcal{L} = l] \tag{B.19}$$

We want $\Pr[\mathsf{RR\text{-}d}(\mathcal{D}) = 1] = \Pr[\mathsf{Submaj}(\mathcal{D}) = 1]$.

If m is odd, for any $l \leq \frac{K-1}{2}$, this is

$$\begin{aligned} \frac{1}{2}(1 - \gamma_{Sub}(l)) &= \sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} \\ \Rightarrow \gamma_{Sub}(l) &= 1 - 2\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} \end{aligned} \tag{B.20}$$

and for any $l \geq \frac{K+1}{2}$, this is

$$\begin{aligned} \frac{1}{2} + \frac{1}{2}\gamma_{Sub}(l) &= \sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} \\ \Rightarrow \gamma_{Sub}(l) &= 2\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - 1 \end{aligned} \tag{B.21}$$

Similarly, if m is even, for any $l \leq \frac{K-1}{2}$, this is

$$\begin{aligned} \frac{1}{2}(1 - \gamma_{Sub}(l)) &= \sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} + \frac{1}{2}\frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} \\ \Rightarrow \gamma_{Sub}(l) &= 1 - 2\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} \end{aligned} \tag{B.22}$$

and for any $l \geq \frac{K+1}{2}$, this is

$$\begin{aligned} \frac{1}{2} + \frac{1}{2}\gamma_{Sub}(l) &= \sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} + \frac{1}{2}\frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} \\ \Rightarrow \gamma_{Sub}(l) &= 2\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} + \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} - 1 \end{aligned} \tag{B.23}$$

Next, we show the above γ_{Sub} is indeed symmetric around $\frac{K}{2}$. For any $l \leq \frac{K-1}{2}$,

there is $K-l \geq \frac{K+1}{2}$. If m is odd,

$$\begin{aligned}\gamma_{Sub}(K-l) &= 2\sum_{j=\frac{m+1}{2}}^{m}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}} - 1 = 2\Big(1-\sum_{j=1}^{\frac{m-1}{2}}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}}\Big) - 1 \\ &= 1 - 2\sum_{j=1}^{\frac{m-1}{2}}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}} = 1 - 2\sum_{j=\frac{m+1}{2}}^{m}\frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} \\ &= \gamma_{Sub}(l) \end{aligned} \tag{B.24}$$

Similarly, if m is even,

$$\begin{aligned}\gamma_{Sub}(K-l) &= 2\sum_{j=\frac{m}{2}+1}^{m}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}} + \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} - 1 = 2\Big(1-\sum_{j=1}^{\frac{m}{2}-1}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}} - \frac{1}{2}\frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}}\Big) - 1 \\ &= 1 - 2\sum_{j=1}^{\frac{m}{2}-1}\frac{\binom{K-l}{j}\binom{l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} = 1 - 2\sum_{j=\frac{m}{2}+1}^{m}\frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} \\ &= \gamma_{Sub}(l) \end{aligned} \tag{B.25}$$

Now, combining Eq. B.20, Eq. B.21 and Eq. B.24, if m is odd, setting γ_{Sub} as

$$\gamma_{Sub}(l) = \gamma_{Sub}(K-l) = 1 - 2\sum_{j=\frac{m+1}{2}}^{m}\frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} \tag{B.26}$$

makes RR-d have the same output distribution as SubMaj.

Similarly, combining Eq. B.22, Eq. B.23 and Eq. B.25, if m is even, setting γ_{Sub} as

$$\gamma_{Sub}(l) = \gamma_{Sub}(K-l) = 1 - 2\sum_{j=\frac{m}{2}+1}^{m}\frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} \tag{B.27}$$

makes RR-d have the same output distribution as SubMaj.

□

B.1.3 Proof of Lemma 3.3.2

Lemma B.1.3 (Restatement of Lemma 3.3.2)**.** *Let $\mathcal{A}$ be an (ϵ, δ)-differentially private algorithm, where $0 \leq \epsilon < c$ for some constant $c > 0$ and $\delta \in [0, \frac{1}{2})$, that computes the majority of K (ϵ, δ)-differentially private mechanisms $M_1, \ldots, M_K$, where $M_i : \mathcal{D} \to \{0, 1\}$ on dataset $\mathcal{D}$ and $\Pr[M_i(\mathcal{D}) = 1] = p_i, \forall i \in [K]$. Then, the error $\mathcal{E}(\mathcal{A}) \geq |\Pr[g(\mathcal{S}) = 1] - \frac{1}{K}\sum_{i=1}^{K} p_i|$, where $g(\mathcal{S})$ is the probability of the true majority output being 1 as defined in Definition 3.1.1.*

Proof. Consider the setting where M_i's are i.i.d., i.e., $\Pr[M_i(\mathcal{D}) = 1] = p, \forall i \in [K]$ for some $p \in [0, 1]$ on any dataset $\mathcal{D}$. Then, it suffices to show $\mathcal{E}(\mathcal{A}) \geq |\Pr[g(\mathcal{S})] = 1 - p|$.

Consider a dataset $\mathcal{D}_0$ such that $\Pr[\mathcal{M}_i(\mathcal{D}_0) = 1] = \Pr[\mathcal{M}_i(\mathcal{D}_0) = 0] = \frac{1}{2}$ and without loss of generality, we may assume $\Pr[\mathcal{A}(\mathcal{D}_0) = 1] \leq \frac{1}{2}$.

Now, we construct a sequence of datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_L$, such that $\mathcal{D}_j$ and $\mathcal{D}_{j+1}$ are neighboring datasets and $\Pr[M_i(\mathcal{D}_j) = 1] = \frac{1}{2}e^{j\epsilon} + \sum_{l=0}^{j-1} e^{l\epsilon}\delta$, $\forall i \in [K]$, $\forall j \in [L]$. Choose $L \in \mathbb{N}$ such that $\frac{1}{2}e^{L\epsilon} + \sum_{l=0}^{L-1} e^{\epsilon l}\delta = p$, for some $p > \frac{1}{2}$.

Now, by definition of differential privacy,

$$
\begin{aligned}
&\Pr[\mathcal{A}(\mathcal{D}_1) = 1] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{D}_0) = 1] + \delta \\
&\Pr[\mathcal{A}(\mathcal{D}_2) = 1] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{D}_1) = 1] + \delta \leq e^{2\epsilon} \Pr[\mathcal{A}(\mathcal{D}_0) = 1] + e^{\epsilon}\delta + \delta \\
&\ldots \\
&\Pr[\mathcal{A}(\mathcal{D}_L) = 1] \leq e^{L\epsilon} \Pr[\mathcal{A}(\mathcal{D}_0) = 1] + \sum_{l=0}^{L-1} e^{\epsilon l}\delta \leq e^{L\epsilon}\frac{1}{2} + \sum_{l=0}^{L-1} e^{\epsilon l}\delta = p
\end{aligned}
$$

Since the probability of true majority being 1 on dataset $\mathcal{D}_L$ is $\Pr[g(\mathcal{S}) = 1] \geq p > \frac{1}{2}$, there is

$$
\mathcal{E}(\mathcal{A}) = |\Pr[g(\mathcal{S}) = 1] - \Pr[\mathcal{A}(\mathcal{D}_L) = 1]| \geq \Pr[g(\mathcal{S}) = 1] - p
$$

□

B.1.4 Proof of Lemma 3.3.3

Lemma B.1.4 (Restatement of Lemma 3.3.3)**.** *Let $\mathcal{A}$ be any randomized algorithm to compute the majority function g on $\mathcal{S}$ such that for all $\mathcal{S}$, $\Pr[\mathcal{A}(\mathcal{S}) = g(\mathcal{S})] \geq 1/2$ (i.e. $\mathcal{A}$ is at least as good as a random guess). Then, there exists a a general function $\gamma : \{0,1\}^{K+1} \to [0,1]$ such that if one sets p_γ by $\gamma(\mathcal{S})$ in DaRRM, the output distribution of DaRRM_γ is the same as the output distribution of $\mathcal{A}$.*

Proof of Lemma 3.3.3. For some $\mathcal{D}$ and conditioned on $\mathcal{S}$, we see that by definition $\Pr[\mathsf{DaRRM}_\gamma(\mathcal{S}) = g(\mathcal{S})] = \gamma(\mathcal{S}) + (1/2)(1 - \gamma(\mathcal{S}))$. We want to set γ such that $\Pr[\mathsf{DaRRM}_\gamma(\mathcal{S}) = g(\mathcal{S})] = \Pr[\mathcal{A}(\mathcal{S}) = g(\mathcal{S})]$. Therefore, we set $\gamma(\mathcal{S}) = 2\Pr[\mathcal{A}(\mathcal{S}) = g(\mathcal{S})] - 1$.

Lastly, we need to justify that $\gamma \in [0,1]$. Clearly, $\gamma(\mathcal{S}) \leq 2 - 1 \leq 1$ since $\Pr[\mathcal{A}(\mathcal{S}) = g(\mathcal{S})] \leq 1$. Note that the non-negativity follows from assumption. □

B.1.5 Proof of Lemma 3.3.4

Lemma B.1.5 (Restatement of Lemma 3.3.4)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, let $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, where $\mathcal{D}$ and $\mathcal{D}'$ are adjacent datasets and $l \in \{0, \ldots, K\}$. For a noise function $\gamma : \{0, 1, \ldots, K\} \to [0,1]$ such that $\gamma(l) = \gamma(K - l), \forall l$, DaRRM_γ is $(m\epsilon, \delta)$-differentially private if and only if for all α_l, α'_l, the following holds,*

$$f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.28}$$

where f is called the ***privacy cost objective*** *and*

$$f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma) := \sum_{l=0}^{\frac{K-1}{2}} (e^{m\epsilon}\alpha'_l - \alpha_l) \cdot \gamma(l) + \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l) \cdot \gamma(l)$$

Proof of Lemma 3.3.4. By the definition of differential privacy (Definition 4.2.1),

DaRRM_γ is $(m\epsilon, \delta)$-differentially private

$\iff \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] \leq e^{m\epsilon}\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1] + \delta,$

$$\text{and } \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 0] \leq e^{m\epsilon} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 0] + \delta, \quad \forall \text{ adjacent datasets } \mathcal{D}, \mathcal{D}' \tag{B.29}$$

Let random variables $\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{K} S(\mathcal{D})$ and $\mathcal{L}(\mathcal{D}') = \sum_{i=1}^{K} S(\mathcal{D}')$ be the sum of observed outcomes on adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, based on which one sets p_γ in DaRRM. Let $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, $\forall l \in \{0, 1, \ldots, K\}$.

Consider the output being 1.

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] \leq e^{m\epsilon} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1] + \delta \tag{B.30}$$

$$\iff \sum_{l=0}^{K} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1 \mid \mathcal{L}(\mathcal{D}) = l] \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.31}$$

$$\leq e^{m\epsilon} \Big(\sum_{l=0}^{K} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1 \mid \mathcal{L}(\mathcal{D}') = l] \cdot \Pr[\mathcal{L}(\mathcal{D}') = l] \Big) + \delta$$

$$\iff \sum_{l=0}^{K} \Big(\gamma(l) \cdot \mathbb{I}\{l \geq \frac{K}{2}\} + \frac{1}{2}(1 - \gamma(l)) \Big) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.32}$$

$$\leq e^{m\epsilon} \Big(\sum_{l=0}^{K} \Big(\gamma(l) \cdot \mathbb{I}\{l \geq \frac{K}{2}\} + \frac{1}{2}(1 - \gamma(l)) \} \Big) \cdot \Pr[\mathcal{L}(\mathcal{D}') = l] \Big) + \delta$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K} \Big(\gamma(l) + \frac{1}{2}(1 - \gamma(l)) \Big) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] + \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}(1 - \gamma(l)) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.33}$$

$$\leq e^{m\epsilon} \Big(\sum_{l=\frac{K+1}{2}}^{K} \Big(\gamma(l) + \frac{1}{2}(1 - \gamma(l)) \Big) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \Big) + e^{m\epsilon} \Big(\sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}(1 - \gamma(l)) \cdot \Pr[\mathcal{L}(\mathcal{D}') = l] \Big) + \delta$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}\gamma(l)\alpha_l - \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}\gamma(l)\alpha_l + \frac{1}{2} \tag{B.34}$$

$$\leq e^{m\epsilon} \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}\gamma(l)\alpha'_l - e^{m\epsilon} \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}\gamma(l)\alpha'_l + \frac{1}{2}e^{m\epsilon} + \delta$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=0}^{\frac{K-1}{2}} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.35}$$

Similarly, consider the output being 0.

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 0] \leq e^{m\epsilon} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 0] + \delta \tag{B.36}$$

$$\iff \sum_{l=0}^{K} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 0 \mid \mathcal{L}(\mathcal{D}) = l] \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.37}$$

$$\leq e^{m\epsilon}\Big(\sum_{l=0}^{K} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 0 \mid \mathcal{L}(\mathcal{D}') = l] \cdot \Pr[\mathcal{L}(\mathcal{D}') = l]\Big) + \delta$$

$$\iff \sum_{l=0}^{K} \Big(\gamma(l) \cdot \mathbb{I}\{l < \frac{K}{2}\} + \frac{1}{2}(1 - \gamma(l))\Big) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.38}$$

$$\leq e^{m\epsilon}\Big(\sum_{l=0}^{K} \gamma(l) \cdot \mathbb{I}\{l < \frac{K}{2}\} + \frac{1}{2}(1 - \gamma(l))\Big) \cdot \Pr[\mathcal{L}(\mathcal{D}') = l] + \delta$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} \Big(\gamma(l) + \frac{1}{2}(1 - \gamma(l))\Big) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] + \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1 - \gamma(l)) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.39}$$

$$\leq e^{m\epsilon}\Big(\sum_{l=0}^{\frac{K-1}{2}} \Big(\gamma(l) + \frac{1}{2}(1 - \gamma(l))\Big) \cdot \Pr[\mathcal{L}(\mathcal{D}') = l] + \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1 - \gamma(l)) \cdot \Pr[\mathcal{L}(\mathcal{D}') = l]\Big) + \delta$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}\gamma(l)\alpha_l - \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}\gamma(l)\alpha_l + \frac{1}{2} \tag{B.40}$$

$$\leq e^{m\epsilon} \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}\gamma(l)\alpha'_l - e^{m\epsilon} \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}\gamma(l)\alpha'_l + \frac{1}{2}e^{m\epsilon} + \delta$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.41}$$

Therefore, plugging Eq. B.35 and Eq. B.41 into Eq. B.29,

DaRRM_γ is $(m\epsilon, \delta)$-differentially private

$$\iff \sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=0}^{\frac{K-1}{2}}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.42}$$

$$\text{and} \sum_{l=0}^{\frac{K-1}{2}}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.43}$$

where $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, $\forall l \in \{0, 1, \ldots, K\}$ and $\mathcal{D}, \mathcal{D}'$ are any adjacent datasets.

Next, we show if γ is symmetric around $\frac{K}{2}$, i.e., $\gamma(l) = \gamma(K - l)$, satisfying either one of Eq. B.42 or Eq. B.43 implies satisfying the other one. Following Eq. B.42,

$$\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=0}^{\frac{K-1}{2}}(\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.44}$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}}(\alpha_{K-l} - e^{m\epsilon}\alpha'_{K-l})\cdot\gamma(K-l) - \sum_{l=\frac{K-1}{2}}^{K}(\alpha_{K-l} - e^{m\epsilon}\alpha'_{K-l})\cdot\gamma(K-l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.45}$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}}(\alpha_{K-l} - e^{m\epsilon}\alpha'_{K-l})\cdot\gamma(l) - \sum_{l=\frac{K-1}{2}}^{K}(\alpha_{K-l} - e^{m\epsilon}\alpha'_{K-l})\cdot\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.46}$$

Since $\gamma(l) = \gamma(K - l)$

For analysis purpose, we rewrite Eq. B.43 as

$$\sum_{l=0}^{\frac{K-1}{2}}(\widetilde{\alpha}_l - e^{m\epsilon}\widetilde{\alpha}'_l)\cdot\gamma(l) - \sum_{l=\frac{K-1}{2}}^{K}(\widetilde{\alpha}_l - e^{m\epsilon}\widetilde{\alpha}'_l)\cdot\gamma(l) \leq e^{m\epsilon} - 1 + 2\delta \tag{B.47}$$

and proceed by showing Eq. B.46 $\iff$ Eq. B.47.

Recall $p_i = \Pr[M_i(\mathcal{D}) = 1]$ and $p'_i = \Pr[M_i(\mathcal{D}') = 1]$. Observe $\mathcal{L}(\mathcal{D}) \sim \text{PoissonBinomial}(\{p_i\}_{i=1}^K)$ and $\mathcal{L}(\mathcal{D}') \sim \text{PoissonBinomial}(\{p'_i\}_{i=1}^K)$. Let $F_l = \{\mathcal{A} : |\mathcal{A}| = l, \mathcal{A} \subseteq [K]\}$, for any $l \in \{0, \ldots, K\}$, denote the set of all subsets of l integers

that can be selected from $[K]$. Let $\mathcal{A}^c = [K] \setminus \mathcal{A}$ be $\mathcal{A}$'s complement set. Notice $F_{K-l} = \{\mathcal{A}^c : \mathcal{A} \in F_l\}$.

Since α denotes the pmf of the Poisson Binomial distribution at l, it follows that

$$\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l] = \sum_{\mathcal{A} \in F_l} \Pi_{i \in \mathcal{A}} p_i \Pi_{j \in \mathcal{A}^c}(1 - p_j) \tag{B.48}$$

Consider $\beta_i = 1 - p_i, \forall i \in [K]$ and a new random variable $\mathcal{L}^\beta \sim \text{PoissonBinomial}(\{\beta_i\}_{i=1}^K)$, and let $\widetilde{\alpha}_l = \Pr[\mathcal{L}^\beta = 1]$. Observe that

$$\begin{aligned}
\widetilde{\alpha}'_l = \Pr[\mathcal{L}^\beta = l] &= \sum_{\mathcal{A} \in F_l} \Pi_{j \in \mathcal{A}} \beta_i \Pi_{i \in \mathcal{A}^c}(1 - \beta_i) = \sum_{\mathcal{A} \in F_l} \Pi_{j \in \mathcal{A}}(1 - p_j) \Pi_{i \in \mathcal{A}^c} p_i \\
&= \sum_{\mathcal{A}^c \in F_{K-l}} \Pi_{j \in \mathcal{A}}(1 - p_i) \Pi_{i \in \mathcal{A}^c} p_i = \sum_{\mathcal{A} \in F_{K-l}} \Pi_{i \in \mathcal{A}} p_i \Pi_{j \in \mathcal{A}^c}(1 - p_i) \\
&= \alpha_{K-l}
\end{aligned} \tag{B.49}$$

Similarly, consider $\beta'_i = 1 - p'_i, \forall i \in [K]$ and a new random variable $\mathcal{L}'^\beta \sim \text{PoissonBinomial}(\beta'_i\}_{i=1}^L)$, and let $\widetilde{\alpha}'_l = \Pr[\mathcal{L}'^\beta = 1]$. Then, $\widetilde{\alpha}'_l = \alpha'_{K-l}$.

Since Eq. B.46 holds for all possible α_{K-l}, α'_{K-l}, Eq. B.47 then holds for all $\widetilde{\alpha}_l, \widetilde{\alpha}'_l$ in the K-simplex, and so Eq. B.47 follows by relabeling α_{K-l} as $\widetilde{\alpha}_l$ and α'_{K-l} as $\widetilde{\alpha}'_l$.

The above implies Eq. B.42 $\iff$ Eq. B.43. Therefore,

$$\begin{aligned}
&\mathsf{DaRRM}_\gamma \text{ is } (m\epsilon, \delta)\text{-differentially private} \\
\iff &\underbrace{\sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=0}^{\frac{K-1}{2}} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l)}_{:= f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma)} \leq e^{m\epsilon} - 1 + 2\delta
\end{aligned} \tag{B.50}$$

□

B.2 Details of Section 3.4: Provable Privacy Amplification

In this section, we consider Problem 3.1.1 in the pure differential privacy and i.i.d. mechanisms setting. That is, $\delta = \Delta = 0$ and $p = p_i = \Pr[M_i(\mathcal{D}) = 1], p' = p'_i = \Pr[M_i(\mathcal{D}') = 1], \forall i \in [K]$. Our goal is to search for a good noise function γ such that: 1) DaRRM_γ is $m\epsilon$-DP, and 2) DaRRM_γ achieves higher utility than that of the baselines (see Section 3.3) under a fixed privacy loss. Our main finding of such a γ function is presented in Theorem 3.4.1, which states given a privacy allowance $m \in [K]$, one can indeed output the majority of $2m-1$ subsampled mechanisms, instead of just m as indicated by simple composition. Later, we formally verify in Lemma B.2.11, Section B.2.3 that taking the majority of more mechanisms strictly increases the utility.

To start, by Lemma 3.3.4, for any noise function γ, γ satisfying goal 1) is equivalent to satisfying

$$f(p, p'; \gamma) \leq e^\epsilon - 1 \tag{B.51}$$

where $f(p, p'; \gamma) = \sum_{l=0}^{\frac{K-1}{2}} (e^{m\epsilon}\alpha'_l - \alpha_l) \cdot \gamma(l) + \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l) \cdot \gamma(l)$ refers to the privacy cost objective (see Lemma 3.3.4) in the i.i.d. mechanisms setting, and recall $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, $\forall l \in \{0, 1, \ldots, K\}$. Notice in this setting, $\mathcal{L}(\mathcal{D}) \sim \text{Binomial}(p)$, and $\mathcal{L}(\mathcal{D}') \sim \text{Binomial}(p')$.

Monotonicity Assumption. For analysis, we restrict our search for a γ function with good utility to the class with a mild monotonicity assumption: $\gamma(l) \geq \gamma(l+1), \forall l \leq \frac{K-1}{2}$ and $\gamma(l) \leq \gamma(l+1), \forall l \geq \frac{K+1}{2}$. This matches our intuition that as $\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{K} S_i$, i.e., the number of mechanisms outputting 1, approaches 0 or K, there is a clearer majority and so not much noise is needed to ensure privacy, which implies a larger value of γ.

Roadmap of Proof of Theorem 3.4.1. Since γ needs to enable Eq. B.51 to be satisfied for all $p, p' \in [0, 1]$, we begin by showing characteristics of **the worst case probabilities**, i.e., $(p^*, p'^*) = \text{argmax}_{(p,p')} f(p, p'; \gamma)$, given any $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ that is symmetric around $\frac{K}{2}$ and that satisfies the above monotonicity assumption,

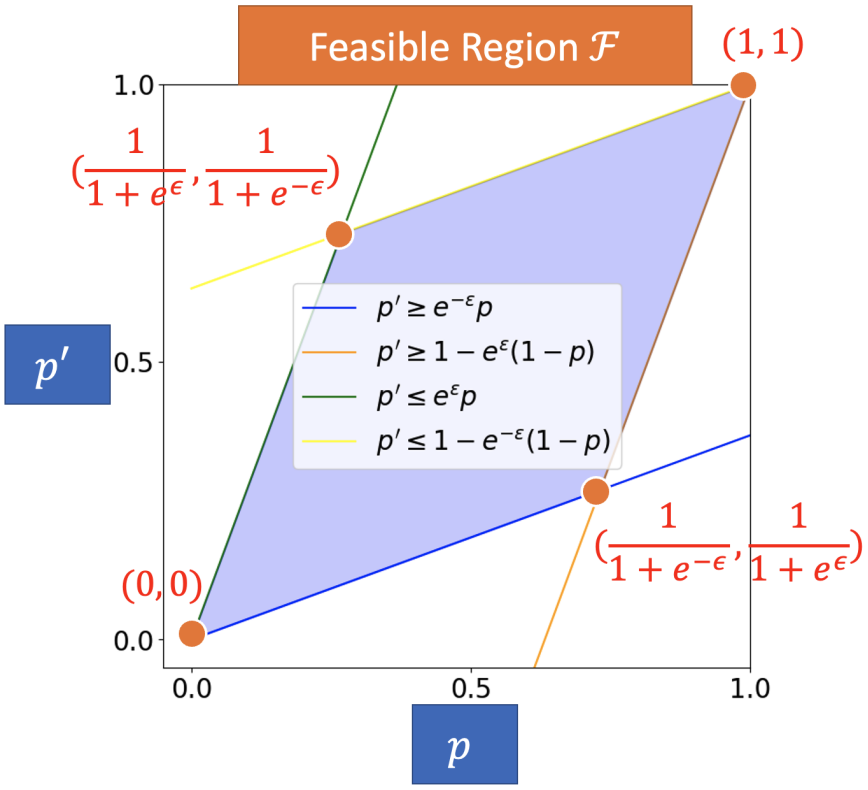

Figure B.2: The feasible region $\mathcal{F}$ is plotted as the blue area. The four boundaries are implied by p, p' satisfying ϵ-differential privacy.

in Lemma B.2.1, Section B.2.1. We call (p^*, p'^*) the worst case probabilities, since they incur the largest privacy loss. Later in Section B.2.2, we present the main proof of Theorem 3.4.1, where we focus on searching for a good γ that enables $f(p^*, p'^*; \gamma) \leq e^{\epsilon} - 1$, based on the characteristics of (p^*, p'^*) in Lemma B.2.1, to ensure DaRRM_{γ} is $m\epsilon$-differentially private.

B.2.1 Characterizing the Worst Case Probabilities

First, note (p, p') are close to each other and lie in a feasible region $\mathcal{F}$, due to each mechanism M_i being ϵ-differentially private; and so does (p^*, p'^*). The feasible region, as illustrated in Figure B.2, is bounded by (a) $p' \leq e^{\epsilon}p$ (b) $p \leq e^{\epsilon}p'$ (c) $1 - p' \leq e^{\epsilon}(1 - p)$, and (d) $1 - p \leq e^{\epsilon}(1 - p')$, where the four boundaries are derived from the definition of differential privacy. Therefore, we only need to search

for $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma)$.

Next, we show that given γ satisfying certain conditions, (p^*, p'^*) can only be on two of the four boundaries of $\mathcal{F}$ in Lemma B.2.1 — that is, either $p^* = e^\epsilon p'$, i.e., on the blue line in Figure B.2, or $1 - p'^* = e^\epsilon(1 - p^*)$, i.e., on the orange line in Figure B.2.

Lemma B.2.1 (Characteristics of worst case probabilities)**.** *For any noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ that is 1) symmetric around $\frac{K}{2}$, 2) satisfies the monotonicity assumption, and 3) $\gamma(\frac{K-1}{2}) > 0$ and $\gamma(\frac{K+1}{2}) > 0$, the worst case probabilities given γ, $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma)$, must satisfy one of the following two equalities:*

$$\begin{aligned} & p^* = e^\epsilon p'^*, && \forall p^* \in [0, \frac{1}{e^{-\epsilon}+1}], p'^* \in [0, \frac{1}{1+e^\epsilon}] \\ \textit{or} \quad & 1 - p'^* = e^\epsilon(1 - p^*), && \forall p^* \in [\frac{1}{1+e^{-\epsilon}}, 1], p'^* \in [\frac{1}{1+e^\epsilon}, 1] \end{aligned}$$

To show Lemma B.2.1, we first show in Lemma B.2.2 that the search of (p^*, p'^*) can be refined to one of the four boundaries of $\mathcal{F}$, via a careful gradient analysis of $f(p, p'; \gamma)$ in $\mathcal{F}$, and then show in Lemma B.2.3 that the search of (p^*, p'^*) can be further refined to two of the four boundaries, due to symmetry of p, p'. Lemma B.2.1 directly follows from the two.

Lemma B.2.2. *For any noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ that is 1) symmetric around $\frac{K}{2}$, 2) satisfies the monotonicity assumption, and 3) $\gamma(\frac{K-1}{2}) > 0$ and $\gamma(\frac{K+1}{2}) > 0$, the worst case probabilities given γ, $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma)$, must satisfy one of the following four equalities:*

$$\begin{aligned} & p'^* = e^\epsilon p^*, && \forall p^* \in [0, \frac{1}{1+e^\epsilon}], p'^* \in [0, \frac{1}{1+e^{-\epsilon}}] \\ & p^* = e^\epsilon p'^*, && \forall p^* \in [0, \frac{1}{e^{-\epsilon}+1}], p'^* \in [0, \frac{1}{1+e^\epsilon}] \\ & 1 - p^* = e^\epsilon(1 - p'^*), && \forall p^* \in [\frac{1}{1+e^\epsilon}, 1], p'^* \in [\frac{1}{1+e^{-\epsilon}}, 1] \\ & 1 - p'^* = e^\epsilon(1 - p^*), && \forall p^* \in [\frac{1}{1+e^{-\epsilon}}, 1], p'^* \in [\frac{1}{1+e^\epsilon}, 1] \end{aligned}$$

Proof of Lemma B.2.2. Recall the privacy cost objective (as defined in Lemma 3.3.4)

is now

$$f(p,p';\gamma)=\sum_{l=0}^{\frac{K-1}{2}}(e^{m\epsilon}\alpha'_l-\alpha_l)\cdot\gamma(l)+\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l-e^{m\epsilon}\alpha'_l)\cdot\gamma(l)$$

where $\alpha_l=\Pr[\mathcal{L}(\mathcal{D})=l]$ and $\alpha'_l=\Pr[\mathcal{L}(\mathcal{D}')=l]$, $\forall l\in\{0,1,\ldots,K\}$. Since $\mathcal{L}(\mathcal{D})\sim\text{Binomial}(p)$ and $\mathcal{L}(\mathcal{D}')\sim\text{Binomial}(p')$ in the i.i.d. mechanisms setting, and using the pmf of the Binomial distribution, f can be written as

$$f(p,p';\gamma)=\sum_{l=0}^{\frac{K-1}{2}}(e^{m\epsilon}\binom{K}{l}p'^l(1-p')^{K-l}-\binom{K}{l}p^l(1-p)^{K-l})\cdot\gamma(l)+\sum_{l=\frac{K+1}{2}}^{K}(\binom{K}{l}p^l(1-p)^{K-l}-e^{m\epsilon}\binom{K}{l}$$

The gradients w.r.t. p and p' are

$$\nabla_p f(p,p';\gamma)=\underbrace{\sum_{l=0}^{\frac{K-1}{2}}-\binom{K}{l}\gamma(l)\cdot(lp^{l-1}(1-p)^{K-l}-p^l(K-l)(1-p)^{K-l-1})}_{:=A}\tag{B.52}$$
$$+\underbrace{\sum_{l=\frac{K+1}{2}}^{K}\binom{K}{l}\gamma(l)\cdot(lp^{l-1}(1-p)^{K-l}-p^l(K-l)(1-p)^{K-l-1})}_{:=B}$$

and

$$\nabla_{p'} f(p,p';\gamma)=\sum_{l=0}^{\frac{K-1}{2}}e^{m\epsilon}\binom{K}{l}\gamma(l)\cdot(lp'^{l-1}(1-p')^{K-l}-p'^l(K-l)(1-p')^{K-l-1})$$
$$+\sum_{l=\frac{K+1}{2}}^{K}-e^{m\epsilon}\binom{K}{l}\gamma(l)\cdot(lp'^{l-1}(1-p')^{K-l}-p'^l(K-l)(1-p')^{K-l-1})\tag{B.53}$$

We show in the following $\forall p\in(0,1)$, $\nabla_p f(p,p';\gamma)>0$ and $\nabla_{p'} f(p,p';\gamma)<0$. This implies there is no local maximum inside $\mathcal{F}$, and so $(p^*,p'^*)=\text{argmax}_{p,p'}\,f(p,p';\gamma)$ must be on one of the four boundaries of $\mathcal{F}$. Also, if $p=0$, then $p'=0$, and $(0,0)$ is a

corner point at the intersection of two boundaries. Similarly, if $p = 1$, then $p' = 1$, and $(1, 1)$ is also a corner point. This concludes $\forall p \in [0, 1]$, $(p^*, p'^*) = \text{argmax}_{p,p'} f(p, p'; \gamma)$ must be on one of the four boundaries of $\mathcal{F}$.

To show $\nabla_p f(p, p'; \gamma) > 0$ for $p \in (0, 1)$, we write $\nabla_p f(p, p'; \gamma) = A + B$ as in Eq. B.52, and show that $A > 0$ and $B > 0$.

To show $A > 0$, first note

$$A := \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K}{l} \cdot (p^l (K-l)(1-p)^{K-l-1} - l p^{l-1}(1-p)^{K-l}) > 0 \tag{B.54}$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K}{l} \cdot p^l (K-l)(1-p)^{K-l-1} > \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K}{l} \cdot l p^{l-1}(1-p)^{K-l}) \tag{B.55}$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K-1}{l} \frac{K}{K-l} \cdot p^l (K-l)(1-p)^{K-l-1} > \sum_{l=1}^{\frac{K-1}{2}} \gamma(l) \binom{K-1}{l-1} \frac{K}{l} \cdot l p^{l-1}(1-p)^{K-l} \tag{B.56}$$

$$\iff K \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K-1}{l} p^l (1-p)^{K-l-1} > K \sum_{l=1}^{\frac{K-1}{2}} \gamma(l) \binom{K-1}{l-1} p^{l-1}(1-p)^{K-l} \tag{B.57}$$

$$\iff \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K-1}{l} p^l (1-p)^{K-l-1} > \sum_{l=0}^{\frac{K-1}{2}-1} \gamma(l+1) \binom{K-1}{l} p^l (1-p)^{K-l-1} \tag{B.58}$$

Since $\forall l \leq \frac{K-1}{2}$, $\gamma(l) \geq \gamma(l+1)$ and $p \in (0, 1)$, there is for $l \in \{0, \ldots, \frac{K-1}{2} - 1\}$,

$$\gamma(l) \binom{K-1}{l} p^l (1-p)^{K-l-1} \geq \gamma(l+1) \binom{K-1}{l} p^l (1-p)^{K-l-1} \tag{B.59}$$

Furthermore, since $\gamma(\frac{K-1}{2}) > 0$ and $p \in (0, 1)$,

$$\gamma(\frac{K-1}{2}) \binom{K-1}{\frac{K-1}{2}} p^{\frac{K-1}{2}} (1-p)^{\frac{K-1}{2}} > 0 \tag{B.60}$$

Eq. B.59 and Eq. B.60 combined implies

$$\gamma(\frac{K-1}{2})\binom{K-1}{\frac{K-1}{2}}p^{\frac{K-1}{2}}(1-p)^{\frac{K-1}{2}}+\sum_{l=0}^{\frac{K-1}{2}-1}\gamma(l)\binom{K-1}{l}p^l(1-p)^{K-l-1}>\sum_{l=0}^{\frac{K-1}{2}-1}\gamma(l+1)\binom{K-1}{l}p^l(1 \tag{B.61}$$

and hence, Eq. B.58 holds. This further implies $A > 0$.

Next, to show $B > 0$, note that

$$B:=\sum_{l=\frac{K+1}{2}}^{K}\binom{K}{l}\gamma(l)\cdot(lp^{l-1}(1-p)^{K-l}-p^l(K-l)(1-p)^{K-l-1})>0 \tag{B.62}$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K}\binom{K}{l}\gamma(l)\cdot lp^{l-1}(1-p)^{K-l}>\sum_{l=\frac{K+1}{2}}^{K}\binom{K}{l}p^l(K-l)(1-p)^{K-l-1} \tag{B.63}$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K}\gamma(l)\binom{K-1}{l-1}\frac{K}{l}\cdot lp^{l-1}(1-p)^{K-l} \tag{B.64}$$

$$>\sum_{l=\frac{K+1}{2}}^{K-1}\gamma(l)\binom{K-1}{l}\frac{K}{K-l}\cdot p^l(K-l)(1-p)^{K-l-1}$$

$$\iff K\sum_{l=\frac{K+1}{2}}^{K}\gamma(l)\binom{K-1}{l-1}\cdot p^{l-1}(1-p)^{K-l} \tag{B.65}$$

$$>K\sum_{l=\frac{K+1}{2}}^{K-1}\gamma(l)\binom{K-1}{l}\cdot p^l(1-p)^{K-l-1}$$

$$\iff \sum_{l=\frac{K+1}{2}}^{K}\gamma(l)\binom{K-1}{l-1}\cdot p^{l-1}(1-p)^{K-l}>\sum_{l=\frac{K+1}{2}+1}^{K}\gamma(l-1)\binom{K-1}{l-1}\cdot p^{l-1}(1-p)^{K-l} \tag{B.66}$$

Since $\forall l \geq \frac{K+1}{2}$, $\gamma(l) \geq \gamma(l-1)$ and $p \in (0,1)$, there is for $l \in \{\frac{K+1}{2}+1,\ldots,K\}$,

$$\gamma(l)\binom{K-1}{l-1}p^{l-1}(1-p)^{K-l}\geq\gamma(l-1)\binom{K-1}{l-1}p^{l-1}(1-p)^{K-l} \tag{B.67}$$

Furthermore, since $\gamma(\frac{K+1}{2}) > 0$ and $p \in (0,1)$,

$$\gamma(\frac{K+1}{2})\binom{K-1}{\frac{K-1}{2}} p^{\frac{K-1}{2}}(1-p)^{\frac{K-1}{2}} > 0 \tag{B.68}$$

Eq. B.67 and Eq. B.68 combined implies

$$\gamma(\frac{K+1}{2})\binom{K-1}{\frac{K-1}{2}} p^{\frac{K-1}{2}}(1-p)^{\frac{K-1}{2}} + \sum_{l=\frac{K+1}{2}+1}^{K} \gamma(l)\binom{K-1}{l-1} \cdot p^{l-1}(1-p)^{K-l} > \sum_{l=\frac{K+1}{2}+1}^{K} \gamma(l-1)\binom{K}{l-} \tag{B.69}$$

and hence Eq. B.66 holds. This further implies $B > 0$.

Following Eq.B.52, for $p \in (0,1)$ and γ satisfying the three assumptions,

$$\nabla_p f(p, p'; \gamma) = A + B > 0 \tag{B.70}$$

Following similar techniques, one can show for $p \in (0,1)$ and γ satisfying the three conditions,

$$\nabla_{p'} f(p, p'; \gamma) < 0 \tag{B.71}$$

This implies there is no local minima or local maxima inside the feasible region $\mathcal{F}$. Also recall $(p, p') \in \{(0,0), (1,1)\}$ are two special cases where (p, p') is at the intersection of two boundaries. Hence, we conclude the worst case probability $(p^*, p'^*) = \text{argmax}_{p,p' \in \mathcal{F}} f(p, p'; \gamma)$ is on one of the four boundaries of $\mathcal{F}$ — that is, (p^*, p'^*) satisfy one of the following:

$$\begin{aligned}
&p'^* = e^{\epsilon} p^*, && \forall p \in [0, \frac{1}{1+e^{\epsilon}}], p' \in [0, \frac{1}{1+e^{-\epsilon}}] \\
&p^* = e^{\epsilon} p'^*, && \forall p \in [0, \frac{1}{e^{-\epsilon}+1}], p' \in [0, \frac{1}{1+e^{\epsilon}}] \\
&1 - p^* = e^{\epsilon}(1 - p'^*), && \forall p \in [\frac{1}{1+e^{\epsilon}}, 1], p' \in [\frac{1}{1+e^{-\epsilon}}, 1] \\
&1 - p'^* = e^{\epsilon}(1 - p^*), && \forall p \in [\frac{1}{1+e^{-\epsilon}}, 1], p' \in [\frac{1}{1+e^{\epsilon}}, 1]
\end{aligned}$$

□

Lemma B.2.3. *For any noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ function that is 1) symmetric around $\frac{K}{2}$ and 2) satisfies the monotonicity assumption, the privacy cost objective $f(p, p'; \gamma)$ is maximized when $p \geq p'$.*

Proof of Lemma B.2.3. Following Eq. B.30 and Eq. B.35 in the proof of Lemma 3.3.4, and that $\delta = 0$,

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] \leq e^{m\epsilon} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1] \tag{B.72}$$

$$\iff \underbrace{\sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l) - \sum_{l=0}^{\frac{K-1}{2}} (\alpha_l - e^{m\epsilon}\alpha'_l)\gamma(l)}_{=f(p,p';\gamma)} \leq e^{m\epsilon} - 1 \tag{B.73}$$

where $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ and $\alpha'_l = \Pr[\mathcal{L}(\mathcal{D}') = l]$, $\forall l \in \{0, 1, \ldots, K\}$. This implies

$$f(p, p'; \gamma) = \frac{\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1]}{\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1]} - 1 \tag{B.74}$$

Hence, $f(p, p'; \gamma)$ is maximized when $\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] \geq \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1]$.

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] = \sum_{l=0}^{K} \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1 \mid \mathcal{L}(\mathcal{D}) = 1] \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.75}$$

$$= \sum_{l=0}^{K} \left(\gamma(l) \cdot \mathbb{I}\{l \geq \frac{K}{2}\} + \frac{1}{2}(1 - \gamma(l))\right) \cdot \Pr[\mathcal{L}(\mathcal{D}) = l] \tag{B.76}$$

$$= \sum_{l=0}^{\frac{K-1}{2}} \frac{1}{2}(1 - \gamma(l)) \cdot \alpha_l + \sum_{l=\frac{K+1}{2}}^{K} \left(\gamma(l) + \frac{1}{2}(1 - \gamma(l))\right) \cdot \alpha_l \tag{B.77}$$

$$= \frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} \gamma(l) \binom{K}{l} p^l (1-p)^{K-l} - \frac{1}{2} \sum_{l=0}^{\frac{K-1}{2}} \gamma(l) \binom{K}{l} p^l (1-p)^{K-l-1} + \frac{1}{2} \tag{B.78}$$

where the last line follows from the observation that in the i.i.d. mechanisms setting, $\mathcal{L}(\mathcal{D}) \sim \text{Binomial}(p)$ and α_l is hence the pmf of the Binomial distribution at l.

Similarly,

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}')=1]=\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}\gamma(l)\binom{K}{l}p'^l(1-p')^{K-l}-\frac{1}{2}\sum_{l=0}^{\frac{K-1}{2}}\gamma(l)\binom{K}{l}p'^l(1-p')^{K-l-1}+\frac{1}{2} \tag{B.79}$$

Now define the objective

$$h(\beta)=\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}\gamma(l)\binom{K}{l}\beta^l(1-\beta)^{K-l}-\frac{1}{2}\sum_{l=0}^{\frac{K-1}{2}}\gamma(l)\binom{K}{l}\beta^l(1-\beta)^{K-l-1}+\frac{1}{2} \tag{B.80}$$

for $\beta\in[0,1]$ and it follows that $\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D})=1]=h(p)$ and $\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}')=1]=h(p')$. We now analyze the monotonicity of $h(\beta)$ in β.

For ease of presentation, define $g(l):=\begin{cases}-\frac{1}{2}\gamma(l) & \forall l\le\frac{K}{2}\\ \frac{1}{2}\gamma(l) & \forall l\ge\frac{K}{2}\end{cases}$. Since $\gamma(l)\ge\gamma(l+1),\forall l\le\frac{K}{2}$ and $\gamma(l+1)\ge\gamma(l),\forall l\ge\frac{K}{2}$, there is $g(l+1)\ge g(l),\forall l\in\{0,\dots,K\}$. And replacing $\gamma(l)$ with $g(l)$ in Eq. B.80,

$$h(\beta)=\sum_{l=0}^{K}g(l)\binom{K}{l}\beta^l(1-\beta)^{K-l} \tag{B.81}$$

$$\nabla_\beta h(\beta)=\sum_{l=0}^{K}g(l)\binom{K}{l}\left(l\beta^{l-1}(1-\beta)^{K-l}-(K-l)\beta^l(1-\beta)^{K-l-1}\right) \tag{B.82}$$

$$=\sum_{l=1}^{K}g(l)\binom{K-1}{l-1}\frac{K}{l}l\beta^{l-1}(1-\beta)^{K-l}-\sum_{l=0}^{K-1}\binom{K-1}{l}\frac{K}{K-l}(K-l)\beta^l(1-\beta)^{K-l-1} \tag{B.83}$$

$$=K\sum_{l=1}^{K}\binom{K-1}{l-1}\beta^{l-1}(1-\beta)^{K-l}-K\sum_{l=0}^{K-1}\binom{K-1}{l}\beta^l(1-\beta)^{K-l-1} \tag{B.84}$$

$$=K\sum_{l=0}^{K-1}g(l+1)\binom{K-1}{l}\beta^l(1-\beta)^{K-l-1}-K\sum_{l=0}^{K-1}g(l)\binom{K-1}{l}\beta^l(1-\beta)^{K-l-1} \tag{B.85}$$

$$= K \sum_{l=0}^{K-1} \Big(g(l+1) - g(l)\Big) \binom{K-1}{l} \beta^l (1-\beta)^{K-l-1} \tag{B.86}$$

Since $g(l+1) \geq g(l)$ and $\binom{K-1}{l}\beta^l(1-\beta)^{K-l-1} \geq 0$, $\nabla_\beta h(\beta) \geq 0$. This implies $h(\beta)$ is monotonically non-decreasing in β and hence,

$$\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] \geq \Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}') = 1] \iff p \geq p' \tag{B.87}$$

Therefore, $f(p, p'; \gamma)$ is maximzied when $p \geq p'$. □

B.2.2 Proof of Privacy Amplification (Theorem 3.4.1)

Theorem B.2.4 (Restatement of Theorem 3.4.1)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, with i.i.d. mechanisms* $\{M_i\}_{i=1}^K$*, i.e.,* $p_i = p$*,* $p_i' = p'$*,* $\forall i \in [K]$*, the privacy allowance* $m \in [K]$ *and* $\delta = \Delta = 0$*. Let the noise function* $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ *be that:*
if $m \geq \frac{K+1}{2}$*,*

$$\gamma(l) = 1$$

and if $m \leq \frac{K-1}{2}$*,*

$$\gamma(l) = \begin{cases} 1 - 2h(l) & \forall l \leq \frac{K-1}{2} \\ 2h(l) - 1 & \forall l \geq \frac{K+1}{2} \end{cases}$$

where $h(l) = \sum_{i=m}^{2m-1} \frac{\binom{l}{i}\binom{K-l}{2m-1-i}}{\binom{K}{2m-1}}$*, then* DaRRM_γ *is* $m\epsilon$*-differentially private.*

Roadmap. Theorem 3.4.1 consists of two parts: γ under a large privacy allowance $m \geq \frac{K+1}{2}$ and γ under a small privacy allowance $m \leq \frac{K-1}{2}$. We first show in Lemma B.2.5, Section B.2.2 that if $m \geq \frac{K+1}{2}$, setting $\gamma = 1$ suffices to ensure DaRRM_γ to be $m\epsilon$-differentially private, and hence one can always output the true majority of K mechanisms. In contrast, simple composition indicates only when $m = K$ can one output the true majority of K mechanisms. Next, we show in Lemma B.2.10, Section B.2.2 that if $m \leq \frac{K-1}{2}$, one can set γ to be γ_{DSub}, which corresponds to outputting the majority of $2m - 1$ subsampled mechanisms (and

hence the name "Double Subsampling", or DSub). In contrast, simple compositon indicates one can only output the majority of m subsampled mechanisms to make sure the output is $m\epsilon$-differentially private. **Theorem 3.4.1 follows directly from combining Lemma B.2.5 and Lemma B.2.10.**

Privacy Amplification Under A Large Privacy Allowance $m \geq \frac{K+1}{2}$

The proof of Lemma B.2.5 is straightforward. We show that given the constant $\gamma_{max}(l) = 1$, if $m \geq \frac{K+1}{2}$, the worst case probabilities are $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma_{max}) = (0, 0)$ and notice that $f(0, 0; \gamma_{max}) = e^{m\epsilon} - 1$, which satisfies the condition in Lemma 3.3.4. Hence, $\mathsf{DaRRM}_{\gamma_{max}}$ is $m\epsilon$-differentially private.

Lemma B.2.5 (Privacy amplification, $m \geq \frac{K+1}{2}$)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, with i.i.d. mechanisms $\{M_i\}_{i=1}^K$, i.e., $p_i = p$, $p'_i = p'$, $\forall i \in [K]$, the privacy allowance $m \geq \frac{K+1}{2}, m \in \mathbb{Z}$ and $\delta = \Delta = 0$. Let the noise function be the constant $\gamma_{max}(l) = 1, \forall l \in \{0, 1, \ldots, K\}$. Then, $\mathsf{DaRRM}_{\gamma_{max}}$ is $m\epsilon$-differentially private.*

Proof of Lemma B.2.5. First, notice $\gamma_{max}(l) = 1, \forall l \in \{0, 1, \ldots, K\}$ is: 1) symmetric around $\frac{K}{2}$, 2) satisfies the monotonicity assumption, and 3) $\gamma_{max}(\frac{K-1}{2}) > 0$ and $\gamma_{max}(\frac{K+1}{2}) > 0$. Therefore, by Lemma B.2.1, the worst case probabilities given γ_{max}, i.e., $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma_{max})$, are on one of the two boundaries of $\mathcal{F}$, satisfying

$$\begin{aligned} & p^* = e^\epsilon p'^*, && \forall p^* \in [0, \frac{1}{e^{-\epsilon}+1}], p'^* \in [0, \frac{1}{1+e^\epsilon}] \\ \text{or} \quad & 1 - p'^* = e^\epsilon(1 - p^*), && \forall p^* \in [\frac{1}{1+e^{-\epsilon}}, 1], p'^* \in [\frac{1}{1+e^\epsilon}, 1] \end{aligned}$$

We now find the local maximums on the two possible boundaries, i.e.,

$$(p^*_{local}, p'^*_{local}) = \underset{(p,p'):p=e^\epsilon p', p\in[0,\frac{1}{e^{-\epsilon}+1}]}{\text{argmax}} f(p, p'; \gamma_{max})$$

and

$$(p^*_{local}, p'^*_{local}) = \underset{(p,p'):1-p'=e^\epsilon(1-p), p\in[\frac{1}{1+e^{-\epsilon}},1]}{\text{argmax}} f(p, p'; \gamma_{max})$$

separately.

Part I: Local worst case probabilities on the boundary $p = e^{\epsilon}p'$.

Plugging $p = e^{\epsilon}p'$ into the privacy cost objective $f(p, p'; \gamma_{max})$, one gets

$$f(p';\gamma_{max}) = \sum_{l=0}^{\frac{K-1}{2}}(e^{m\epsilon}\binom{K}{l}p'^{l}(1-p')^{K-l} - \binom{K}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l}) \tag{B.88}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K}(\binom{K}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l} - e^{m\epsilon}\binom{K}{l}p'^{l}(1-p')^{K-l})$$

The gradient w.r.t. p' is

$$\nabla_{p'} f(p';\gamma_{max}) = \sum_{l=0}^{\frac{K-1}{2}}\Big(e^{m\epsilon}\binom{K}{l}(lp'^{l-1}(1-p')^{K-l} - p'^{l}(K-l)(1-p')^{K-l-1}) \tag{B.89}$$
$$- e^{\epsilon}\binom{K}{l}(l(e^{\epsilon}p')^{l-1}(1-e^{\epsilon}p')^{K-l} - e^{\epsilon l}p'^{l}(K-l)(1-e^{\epsilon}p')^{K-l-1})\Big)$$
$$+ \sum_{l=\frac{K+1}{2}}^{K}\Big(e^{\epsilon}\binom{K}{l}(l(e^{\epsilon}p')^{l-1}(1-e^{\epsilon}p')^{K-l} - e^{\epsilon l}p'^{l}(K-l)(1-e^{\epsilon}p')^{K-l-1})$$
$$- e^{m\epsilon}\binom{K}{l}(lp'^{l-1}(1-p')^{K-l} - p'^{l}(K-l)(1-p')^{K-l-1})\Big)$$
$$= -K\sum_{l=0}^{\frac{K-1}{2}}e^{m\epsilon}\binom{K-1}{l}p'^{l}(1-p')^{K-l-1} + K\sum_{l=\frac{K+1}{2}}^{K-1}e^{m\epsilon}\binom{K-1}{l}p'^{l}(1-p')^{K-l-1} \tag{B.90}$$
$$+ K\sum_{l=0}^{\frac{K-1}{2}}e^{\epsilon}\binom{K-1}{l}(\epsilon p')^{\epsilon}(1-e^{\epsilon}p')^{K-l-1} - K\sum_{l=\frac{K+1}{2}}^{K-1}e^{\epsilon}\binom{K-1}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1}$$
$$+ K\sum_{l=0}^{\frac{K-1}{2}-1}e^{m\epsilon}\binom{K-1}{l}p'^{l}(1-p')^{K-l-1} - K\sum_{l=\frac{K-1}{2}}^{K-1}e^{m\epsilon}\binom{K-1}{l}p'^{l}(1-p')^{K-l-1}$$
$$- K\sum_{l=0}^{\frac{K-1}{2}-1}e^{\epsilon}\binom{K-1}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1} + K\sum_{l=\frac{K-1}{2}}^{K-1}e^{\epsilon}\binom{K-1}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1}$$

$$= -2K \underbrace{e^{m\epsilon}\binom{K-1}{\frac{K-1}{2}} p'^{\frac{K-1}{2}}(1-p')^{\frac{K-1}{2}}}_{:=A} + 2K \underbrace{e^{\epsilon}\binom{K-1}{\frac{K-1}{2}}(e^{\epsilon}p')^{\frac{K-1}{2}}(1-e^{\epsilon}p')^{\frac{K-1}{2}}}_{:=B} \tag{B.91}$$

Notice that

$$\frac{A}{B} = \frac{e^{m\epsilon}\binom{K-1}{\frac{K-1}{2}} p'^{\frac{K-1}{2}}(1-p')^{\frac{K-1}{2}}}{e^{\epsilon}\binom{K-1}{\frac{K-1}{2}}(e^{\epsilon}p')^{\frac{K-1}{2}}(1-e^{\epsilon}p')^{\frac{K-1}{2}}} = \frac{e^{m\epsilon}}{e^{\frac{K+1}{2}\epsilon}} \cdot \left(\frac{1-p'}{1-e^{\epsilon}p'}\right)^{\frac{K-1}{2}} \tag{B.92}$$

Since $\frac{1-p'}{1-e^{\epsilon}p'} \geq 1$ and $m \geq \frac{K+1}{2}$, $\frac{A}{B} \geq 1$. This implies $\nabla_{p'} f(p'; \gamma_{max}) \leq 0$. Hence, $f(p'; \gamma_{max})$ is monotonically non-increasing on the boundary, for $p' \in [0, \frac{1}{1+e^{\epsilon}}]$.

Therefore, $\operatorname{argmax}_{p':p' \in [0, \frac{1}{1+e^{\epsilon}}]} f(p'; \gamma_{max}) = 0$. Since $p = e^{\epsilon}p'$, $p' = 0$ implies $p = 0$.

Hence,

$$(p^*_{local}, p'^*_{local}) = \operatorname*{argmax}_{(p,p'):p=e^{\epsilon}p', p \in [0, \frac{1}{e^{-\epsilon}+1}]} f(p, p'; \gamma_{max}) = (0, 0)$$

and

$$\max_{(p,p'):p=e^{\epsilon}p', p \in [0, \frac{1}{e^{-\epsilon}+1}]} f(p, p'; \gamma_{max}) = f(0, 0; \gamma_{max}) = e^{m\epsilon} - 1$$

Part II: Local worst case probabilities on the boundary $1 - p' = e^{\epsilon}(1 - p)$.

For simplicity, let $q = 1 - p$ and $q' = 1 - p'$. Note on this boundary $p \in [\frac{1}{1+e^{-\epsilon}}, 1]$ and $p' \in [\frac{1}{1+e^{\epsilon}}, 1]$, and hence, $q \in [0, \frac{1}{1+e^{\epsilon}}]$ and $q' \in [0, \frac{1}{1+e^{-\epsilon}}]$.

Plugging q and q' into the privacy cost objective $f(p, p'; \gamma_{max})$, one gets a new objective in q, q' as

$$\begin{aligned} f(q, q'; \gamma_{max}) = &\sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon}\binom{K}{l}(1-q')^l q'^{K-l} - \binom{K}{l}(1-q)^l q^{K-l} \right) \cdot \gamma_{max}(l) \\ &+ \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l}(1-q)^l q^{K-l} - e^{m\epsilon}\binom{K}{l}(1-q')^l q'^{K-l} \right) \cdot \gamma_{max}(l) \end{aligned} \tag{B.93}$$

$$= \sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon} \binom{K}{l} (1-q')^l q'^{K-l} - \binom{K}{l} (1-q)^l q^{K-l} \right) \tag{B.94}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l} (1-q)^l q^{K-l} - e^{m\epsilon} \binom{K}{l} (1-q')^l q'^{K-l} \right)$$

Since on this boundary, $1-p' = e^\epsilon(1-p)$, writing this in q, q', this becomes $q' = e^\epsilon q$. Plugging $q' = e^\epsilon q$ into $f(q, q'; \gamma_{max})$, one gets

$$f(q;\gamma_{max}) = \sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon} \binom{K}{l} (1-e^\epsilon q)^l (e^\epsilon q)^{K-l} - \binom{K}{l} (1-q)^l q^{K-l} \right) \tag{B.95}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l} (1-q)^l q^{K-l} - e^{m\epsilon} \binom{K}{l} (1-e^\epsilon q)^l (e^\epsilon q)^{K-l} \right)$$

The gradient w.r.t. q is

$$\nabla_q f(q) = \sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon} \binom{K}{l} \left((-e^\epsilon) l (1-e^\epsilon q)^{l-1} (e^\epsilon q)^{K-l} + e^\epsilon (K-l)(1-e^\epsilon q)^l (e^\epsilon q)^{K-l-1} \right) \right. \tag{B.96}$$
$$\left. - \binom{K}{l} \Big(-l(1-q)^{l-1} q^{K-l} + (K-l)(1-q)^l q^{K-l-1} \Big) \right)$$
$$+ \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l} \Big(-l(1-q)^{l-1} q^{K-l} + (K-l)(1-q)^l q^{K-l-1} \Big) \right.$$
$$\left. - e^{m\epsilon} \binom{K}{l} \left((-e^\epsilon) l (1-e^\epsilon q)^{l-1} (e^\epsilon q)^{K-l} + e^\epsilon (K-l)(1-e^\epsilon q)^l (e^\epsilon q)^{K-l-1} \right) \right)$$
$$= -\sum_{l=1}^{\frac{K-1}{2}} e^{(m+1)\epsilon} \binom{K-1}{l-1} \frac{K}{l} l (1-e^\epsilon q)^{l-1} (e^\epsilon q)^{K-l} + \sum_{l=0}^{\frac{K-1}{2}} e^{(m+1)\epsilon} \binom{K-1}{l} \frac{K}{K-l} (K-l)(1-e^\epsilon q)^l \tag{B.97}$$
$$+ \sum_{l=1}^{\frac{K-1}{2}} \binom{K-1}{l-1} \frac{K}{l} l (1-q)^{l-1} q^{K-l} - \sum_{l=0}^{\frac{K-1}{2}} \binom{K-1}{l} \frac{K}{K-l} (K-l)(1-q)^l q^{K-l-1}$$

$$
\begin{aligned}
&- \sum_{l=\frac{K+1}{2}}^{K} \binom{K-1}{l-1} \frac{K}{l} l(1-q)^{l-1} q^{K-l} + \sum_{l=\frac{K+1}{2}}^{K-1} \binom{K-1}{l} \frac{K}{K-l} (K-l)(1-q)^{l} q^{K-l-1} \\
&+ \sum_{l=\frac{K+1}{2}}^{K} e^{(m+1)\epsilon} \binom{K-1}{l-1} \frac{K}{l} l(1-e^{\epsilon}q)^{l-1} (e^{\epsilon}q)^{K-l} - \sum_{l=\frac{K+1}{2}}^{K-1} e^{(m+1)\epsilon} \binom{K-1}{l} \frac{K}{K-l} (K-l)(1- \\
&= -K \sum_{l=1}^{\frac{K-1}{2}} e^{(m+1)\epsilon} \binom{K-1}{l-1} (1-e^{\epsilon}q)^{l-1} (e^{\epsilon}q)^{K-l} + K \sum_{l=0}^{\frac{K-1}{2}} e^{(m+1)\epsilon} \binom{K-1}{l} (1-e^{\epsilon}q)^{l} (e^{\epsilon}q)^{K-l-}
\end{aligned}
\tag{B.98}
$$

$$
\begin{aligned}
&+ K \sum_{l=1}^{\frac{K-1}{2}} \binom{K-1}{l-1} (1-q)^{l-1} q^{K-l} - K \sum_{l=0}^{\frac{K-1}{2}} \binom{K-1}{l} (1-q)^{l} q^{K-l-1} \\
&- K \sum_{l=\frac{K+1}{2}}^{K} \binom{K-1}{l-1} (1-q)^{l-1} q^{K-l} + K \sum_{l=\frac{K+1}{2}}^{K-1} \binom{K-1}{l} (1-q)^{l} q^{K-l-1} \\
&+ K \sum_{l=\frac{K+1}{2}}^{K} e^{(m+1)\epsilon} \binom{K-1}{l-1} (1-e^{\epsilon}q)^{l-1} (e^{\epsilon}q)^{K-l} - K \sum_{l=\frac{K+1}{2}}^{K-1} e^{(m+1)\epsilon} \binom{K-1}{l} (1-e^{\epsilon}q)^{l} (e^{\epsilon}q)^{K-} \\
&= 2K e^{(m+1)\epsilon} \binom{K-1}{\frac{K-1}{2}} (1-e^{\epsilon}q)^{\frac{K-1}{2}} (e^{\epsilon}q)^{\frac{K-1}{2}} - 2K \binom{K-1}{\frac{K-1}{2}} (1-q)^{\frac{K-1}{2}} q^{\frac{K-1}{2}}
\end{aligned}
\tag{B.99}
$$

Recall $q \in [0, \frac{1}{1+e^{\epsilon}}]$ and so $(1-e^{\epsilon}q)(e^{\epsilon}q) \geq (1-q)q$. Furthermore, since $e^{(m+1)\epsilon} \geq 1$, there is $\nabla_q f(q) \geq 0$. This implies $f(q)$ is monotonically non-decreasing in q, and so the local maximum on this boundary is

$$
(q^*_{local}, q'^*_{local}) = \underset{(q,q'):q'=e^{\epsilon}q, q\in[0,\frac{1}{1+e^{\epsilon}}]}{\operatorname{argmax}} f(q, q'; \gamma_{max}) = (\frac{1}{1+e^{\epsilon}}, \frac{1}{1+e^{-\epsilon}}) \tag{B.100}
$$

That is,

$$
(p^*_{local}, p'^*_{local}) = \underset{(p,p'):1-p'=e^{\epsilon}(1-p), p\in[\frac{1}{1+e^{-\epsilon}},1]}{\operatorname{argmax}} f(p, p'; \gamma_{max}) = (1-q^*_{local}, 1-q'^*_{local}) = (\frac{1}{1+e^{-\epsilon}}, \frac{1}{1+e^{\epsilon}}) \tag{B.101}
$$

Part III: The global worst case probabilities.

Notice that $(\frac{1}{1+e^{-\epsilon}}, \frac{1}{1+e^{\epsilon}})$, the maximum on the second boundary $1-p' = e^{\epsilon}(1-p), \forall p \in [\frac{1}{1+e^{-\epsilon}}, 1]$, is indeed the minimum on the first boundary $p = e^{\epsilon}p', \forall p \in [0, \frac{1}{1+e^{-\epsilon}+1}]$.

Therefore, the global maximum given γ_{max} is

$$(p^*, p'^*) = \underset{(p,p')\in\mathcal{F}}{\operatorname{argmax}} f(p, p'; \gamma_{max}) = \underset{(p,p'):p=e^{\epsilon}p', p\in[0,\frac{1}{1+e^{-\epsilon}}]}{\operatorname{argmax}} f(p, p'; \gamma_{max}) = (0, 0) \quad \text{(B.102)}$$

and recall that $f(0, 0; \gamma_{max}) = e^{m\epsilon} - 1$.

Hence, if $m \geq \frac{K+1}{2}$, by Lemma 3.3.4 $\mathsf{DaRRM}_{\gamma_{max}}$ is $m\epsilon$-differentially private.

□

Privacy Amplification Under A Small Privacy Allowance $m \leq \frac{K-1}{2}$

The proof of Lemma B.2.10 is slightly more involved. First, recall by Lemma 3.3.1, γ_{Sub}, the noise function that makes the output of $\mathsf{DaRRM}_{\gamma_{Sub}}$ and the subsampling baseline the same, is

$$\begin{aligned}\gamma_{Sub}(l) &= \gamma_{Sub}(K = l) \\ &= \begin{cases} 1 - 2\sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} & \text{if } m \text{ is odd} \\ 1 - 2\sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} & \text{if } m \text{ is even} \end{cases}\end{aligned}$$

for $l \in \{0, 1, \ldots, K\}$, suppose the privacy allowance $m \in \mathbb{Z}$.

If we define $h(l) := \begin{cases} \sum_{j=\frac{m+1}{2}}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} & \text{if } m \text{ is odd} \\ \sum_{j=\frac{m}{2}+1}^{m} \frac{\binom{l}{j}\binom{K-l}{m-j}}{\binom{K}{m}} - \frac{\binom{l}{\frac{m}{2}}\binom{K-l}{\frac{m}{2}}}{\binom{K}{m}} & \text{if } m \text{ is even} \end{cases}$, then $\gamma_{Sub}(l)$ can be written as $\gamma_{Sub}(l) = \begin{cases} 1 - 2h(l) & \text{if } l \leq \frac{K-1}{2} \\ 2h(l) - 1 & \text{if } l \geq \frac{K+1}{2} \end{cases}$.

This can be generalized to a broader class of γ functions — which we call the "symmetric form family" — as follows

Definition B.2.6. *$\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ is a member of the "symmetric form*

family" if γ *follows*

$$\gamma(l) = \begin{cases} 1 - 2h(l) & \text{if } l \leq \frac{K-1}{2} \\ 2h(l) - 1 & \text{if } l \geq \frac{K+1}{2} \end{cases} \tag{B.103}$$

where $h : \{0, 1, \ldots, K\} \to [0, 1]$ *and*

$$h(l) + h(K - l) = 1, \quad h(l + 1) \geq h(l), \quad \forall l \in \{0, 1, \ldots, K\}, \quad \text{and} \quad \gamma(\frac{K-1}{2}) > 0, \gamma(\frac{K+1}{2}) > 0$$

It is easy to verify any γ function that belongs to the "symmetric form family" satisfies: 1) symmetric around $\frac{K}{2}$ and 2) the monotonicity assumption. Hence, Lemma B.2.1 can be invoked to find the worst case probabilities given such γ, i.e., $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma)$, which in turn gives us the guarantee of DaRRM_γ being $m\epsilon$-differentially private.

Roadmap. In this section, we restrict our search of a good γ that maximizes the utility of DaRRM_γ to in the "symmetric form family". To show the main privacy amplification result under a small m in Lemma B.2.10, Section B.2.2, we need a few building blocks, shown in Section B.2.2. We first show in Lemma B.2.7, Section B.2.2 two clean sufficient conditions that if a "symmetric form family" γ satisfies, then DaRRM_γ is $m\epsilon$-differentially private, in terms of the expectation of the γ function applied to Binomial random variables. The Binomial random variables appear in the lemma, because recall the sum of the observed outcomes on a dataset $\mathcal{D}$, $\mathcal{L}(\mathcal{D})$, follows a Binomial distribution in the i.i.d. mechanisms setting. Next, we show a recurrence relationship that connects the expectation of Binomial random variables to Hypergeometric random variables in Lemma B.2.9. This is needed because observe that for γ functions that makes DaRRM_γ have the same output as the majority of subsampled mechanisms, the h function is now a sum of pmfs of the Hypergeometric random variable.

Finally, the proof of the main result under a small m (Lemma B.2.10) is presented in Section B.2.2, based on Lemma B.2.7 and Lemma B.2.9. We show in Lemma B.2.10 that γ_{DSub}, i.e., the γ function that enables the output of $\mathsf{DaRRM}_{\gamma_{DSub}}$ and outputting the majority of $2m - 1$ subsampled mechanisms to be the same, belongs to the "symmetric form family" and satisfies the sufficient conditions as stated in Lemma B.2.7,

implying $\mathsf{DaRRM}_{\gamma_{DSub}}$ being $m\epsilon$-differentially private.

Building Blocks

Lemma B.2.7 (Privacy conditions of the "symmetric form family" functions)**.** *Let random variables* $X \sim$ *Binomial*$(K-1, p')$, $Y \sim$ *Binomial*$(K-1, e^{\epsilon}p')$, $\hat{X} \sim$ *Binomial*$(K-1, 1-e^{\epsilon}(1-p))$ *and* $\hat{Y} \sim$ *Binomial*$(K-1, p)$. *For a function* $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$ *that belongs to the "symmetric form family" (Definition B.2.6), if* γ *also satisfies both conditions as follows:*

$$e^{m\epsilon}\mathbb{E}_X[h(X+1) - h(X)] \geq e^{\epsilon}\mathbb{E}_Y[h(Y+1) - h(Y)], \quad \forall p' \in [0, \frac{1}{1+e^{\epsilon}}] \tag{B.104}$$

$$e^{(m+1)\epsilon}\mathbb{E}_{\hat{X}}[h(\hat{X}+1) - h(\hat{X})] \geq \mathbb{E}_{\hat{Y}}[h(\hat{Y}+1) - h(\hat{Y})], \quad \forall p \in [\frac{1}{1+e^{-\epsilon}}, 1] \tag{B.105}$$

then Algorithm DaRRM_{γ} *is* $m\epsilon$*-differentially private.*

Proof of Lemma B.2.7. Since $h(l+1) \geq h(l)$ on $l \in \{0, \ldots, K\}$, $\gamma(l) \geq \gamma(l+1), \forall l \leq \frac{K}{2}$ and $\gamma(l+1) \geq \gamma(l), \forall l \geq \frac{K}{2}$. Furthermore, since $h(l) + h(K-l) = 1$, $\gamma(\frac{K-1}{2}) = 1 - 2h(\frac{K-1}{2}) = 1 - 2(1 - h(\frac{K+1}{2})) = 2h(\frac{K+1}{2}) - 1$. Hence, any γ that belongs to the "symmetric form family" satisfies: 1) symmetric around $\frac{K}{2}$, 2) the monotonicity assumption, and 3) $\gamma(\frac{K-1}{2}) = \gamma(\frac{K+1}{2}) > 0$.

Therefore, by Lemma B.2.1, the worst case probabilities $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p, p'; \gamma)$ are on one of the two boundaries of $\mathcal{F}$, satisfying

$$p^* = e^{\epsilon}p'^*, \qquad \forall p^* \in [0, \frac{1}{e^{-\epsilon}+1}], p'^* \in [0, \frac{1}{1+e^{\epsilon}}] \tag{B.106}$$

$$\text{or} \quad 1 - p'^* = e^{\epsilon}(1-p^*), \qquad \forall p^* \in [\frac{1}{1+e^{-\epsilon}}, 1], p'^* \in [\frac{1}{1+e^{\epsilon}}, 1] \tag{B.107}$$

We now derive the sufficient conditions that if any γ from the "symmetric form family" satisfy, then DaRRM_{γ} is $m\epsilon$-differentially private, from the two boundaries as in Eq. B.106 and Eq. B.107 separately.

Part I: Deriving a sufficient condition from Eq. B.106 for "symmetric form family" γ.

Consider the boundary of $\mathcal{F}$, $p = e^{\epsilon}p'$, $\forall p \in [0, \frac{1}{1+e^{-\epsilon}}], p' \in [0, \frac{1}{1+e^{\epsilon}}]$.

Given any γ, plugging $p = e^{\epsilon}p'$ into the privacy cost objective $f(p, p'; \gamma)$, one gets

$$f(p';\gamma) = \sum_{l=0}^{\frac{K-1}{2}} (e^{m\epsilon}\binom{K}{l}p'^{l}(1-p')^{K-l} - \binom{K}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l}) \cdot \gamma(l) \tag{B.108}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K} (\binom{K}{l}(e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l} - e^{m\epsilon}\binom{K}{l}p'^{l}(1-p')^{K-l}) \cdot \gamma(l)$$

The gradient w.r.t. p' is

$$\frac{\nabla_{p'} f(p');\gamma}{K} = e^{m\epsilon} \sum_{l=0}^{\frac{K-1}{2}-1} \binom{K-1}{l} p'^{l}(1-p')^{K-l-1}\Big(\gamma(l+1) - \gamma(l)\Big) - 2e^{m\epsilon}\binom{K-1}{\frac{K-1}{2}}p'^{\frac{K-1}{2}}(1-p')^{\frac{K-} \tag{B.109}$$
$$+ e^{m\epsilon} \sum_{l=\frac{K+1}{2}}^{K-1} \binom{K-1}{l} p'^{l}(1-p')^{K-l-1}\Big(\gamma(l) - \gamma(l+1)\Big)$$
$$+ e^{\epsilon} \sum_{l=0}^{\frac{K-1}{2}-1} \binom{K-1}{l} (e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1}\Big(\gamma(l) - \gamma(l+1)\Big) + 2e^{\epsilon}\binom{K-1}{\frac{K-1}{2}}(e^{\epsilon}p')^{\frac{K-1}{2}}(1-e^{\epsilon}p')^{\frac{K-1}{2}}\gamma($$
$$+ e^{\epsilon} \sum_{l=\frac{K+1}{2}}^{K-1} \binom{K-1}{l} (e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1}\Big(\gamma(l+1) - \gamma(l)\Big)$$

Consider $l \in \{0, 1, \ldots, K\}$ in the above Eq. B.109. For any function γ that belongs to the "symmetric form family",

1. If $l \leq \frac{K}{2}$, $\gamma(l) - \gamma(l+1) = (1 - 2h(l)) - (1 - 2h(l+1)) = 2h(l+1) - 2h(l)$
2. If $l \geq \frac{K}{2}$, $\gamma(l+1) - \gamma(l) = (2h(l+1) - 1) - (2h(l) - 1) = 2h(l+1) - 2h(l)$
3. Since $\gamma(\frac{K-1}{2}) = \gamma(\frac{K+1}{2})$,

$$2\gamma(\frac{K-1}{2}) = \Big(\gamma(\frac{K-1}{2}) + \gamma(\frac{K+1}{2})\Big) \tag{B.110}$$
$$= \Big(1 - 2h(\frac{K-1}{2}) + 2h(\frac{K+1}{2}) - 1\Big) \tag{B.111}$$
$$= 2h(\frac{K+1}{2}) - 2h(\frac{K-1}{2}) \tag{B.112}$$

Hence, following Eq. B.109, the gradient, $\nabla_{p'} f(p';\gamma)$, given a "symmetric form family" γ can be written as

$$\frac{\nabla_{p'} f(p';\gamma)}{K} = -e^{m\epsilon} \sum_{l=0}^{K-1} \binom{K-1}{l} p'^{l}(1-p')^{K-l}\Big(2h(l+1) - 2h(l)\Big) \tag{B.113}$$

$$+ e^{\epsilon} \sum_{l=0}^{K-1} \binom{K-1}{l} (e^{\epsilon}p')^{l}(1-e^{\epsilon}p')^{K-l-1}\Big(2h(l+1) - 2h(l)\Big)$$

$$= -2e^{m\epsilon}\mathbb{E}_X[h(X+1) - h(X)] + 2e^{\epsilon}\mathbb{E}_Y[h(Y+1) - h(Y)] \tag{B.114}$$

where $X \sim \text{Binomial}(K-1, p')$ and $Y \sim \text{Binomial}(K-1, e^{\epsilon}p')$. The above implies

$$\nabla_{p'} f(p';\gamma) \leq 0 \iff e^{\epsilon}\mathbb{E}_Y[h(Y+1) - h(Y)] \leq e^{m\epsilon}\mathbb{E}_X[h(X+1) - h(X)] \tag{B.115}$$

If $\nabla_{p'} f(p';\gamma) \leq 0$, then we know the local worst case probabilities on the boundary $p = e^{\epsilon}p', \forall p \in [0, \frac{1}{1+e^{-\epsilon}}]$ given any γ is $(p^*_{local}, p'^*_{local}) = \text{argmax}_{(p,p'):p=e^{\epsilon}p', p\in[0,\frac{1}{1+e^{-\epsilon}}]} f(p,p';\gamma) = (0,0)$. Furthermore, recall the privacy cost objective given any γ is

$$f(p,p';\gamma)$$
$$= \sum_{l=0}^{\frac{K-1}{2}} (e^{m\epsilon}\alpha'_l - \alpha_l)\cdot\gamma(l) + \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha'_l)\cdot\gamma(l)$$
$$= \sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon}\binom{K}{l}p'^{l}(1-p')^{K-l} - \binom{K}{l}p^{l}(1-p)^{K-l}\right)\cdot\gamma(l) + \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l}p^{l}(1-p)^{K-l} - e^{m\epsilon}\binom{K}{l}p'^{l}(1\right.$$

and so for any γ,

$$f(0,0;\gamma) = (e^{m\epsilon} - 1)\cdot\gamma(0) \leq e^{m\epsilon} - 1 \tag{B.116}$$

Also, notice the local minimum on this boundary is

$$(p_{min}, p'_{min}) = \underset{(p,p'):p=e^{\epsilon}p', p\in[0,\frac{1}{1+e^{-\epsilon}}]}{\text{argmin}} f(p,p'l;\gamma) = (\frac{1}{1+e^{-\epsilon}}, \frac{1}{1+e^{\epsilon}}) \tag{B.117}$$

Part II: Deriving a sufficient condition from Eq. B.107 for "symmetric

form family" γ.

Consider the boundary of $\mathcal{F}$, $1-p' = e^{\epsilon}(1-p)$, $\forall p \in [\frac{1}{1+e^{-\epsilon}}, 1], p' \in [\frac{1}{1+e^{\epsilon}}, 1]$. For simplicity, let $q = 1-p \in [0, \frac{1}{1+e^{\epsilon}}]$ and $q' = 1-p' \in [0, \frac{1}{1+e^{-\epsilon}}]$. Plugging $q' = e^{\epsilon}q$ into the privacy cost objective, one gets, given any γ,

$$f(q;\gamma) = \sum_{l=0}^{\frac{K-1}{2}} \left(e^{m\epsilon} \binom{K}{l} (1-e^{\epsilon}q)^l (e^{\epsilon}q)^{K-l} - \binom{K}{l}(1-q)^l q^{K-l} \right) \cdot \gamma(l) \tag{B.118}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K} \left(\binom{K}{l}(1-q)^l q^{K-l} - e^{m\epsilon}\binom{K}{l}(1-e^{\epsilon}q)^l (e^{\epsilon}q)^{K-l} \right) \cdot \gamma(l)$$

The gradient w.r.t. q is

$$\frac{\nabla_q f(q;\gamma)}{K} = \sum_{l=0}^{\frac{K-1}{2}-1} e^{(m+1)\epsilon} \binom{K-1}{l} (1-e^{\epsilon}q)^l (e^{\epsilon}q)^{K-l-1} \cdot \Big(\gamma(l) - \gamma(l+1)\Big) \tag{B.119}$$
$$+ \sum_{l=\frac{K+1}{2}}^{K-1} \binom{K-1}{l} (1-e^{\epsilon}q)^l (e^{\epsilon}q)^{K-l-1} \cdot \Big(\gamma(l+1) - \gamma(l)\Big) + 2e^{(m+1)\epsilon} \binom{K-1}{\frac{K-1}{2}} (1-e^{\epsilon}q)^{\frac{K}{}}$$
$$+ \sum_{l=0}^{\frac{K-1}{2}-1} \binom{K-1}{l} (1-q)^l q^{K-l-1} \cdot \Big(\gamma(l+1) - \gamma(l)\Big)$$
$$+ \sum_{l=\frac{K+1}{2}}^{K-1} (1-q)^l q^{K-l-1} \cdot \Big(\gamma(l) - \gamma(l+1)\Big) - 2\binom{K-1}{\frac{K-1}{2}} (1-q)^{\frac{K-1}{2}} q^{\frac{K-1}{2}} \cdot \gamma(\frac{K-1}{2})$$

For any function γ that belongs to the "symmetric form family", the gradient $\nabla_q f(q;\gamma)$ can be written as

$$\frac{\nabla_q f(q;\gamma)}{K} = e^{(m+1)\epsilon} \sum_{l=0}^{K-1} \binom{K-1}{l} (1-e^{\epsilon}q)^l (e^{\epsilon}q)^{K-l-1} \cdot \Big(2h(l+1) - 2h(l)\Big) \tag{B.120}$$
$$- \sum_{l=0}^{K} \binom{K-1}{l} (1-q)^l q^{K-l-1} \cdot \Big(2h(l+1) - 2h(l)\Big)$$
$$= 2e^{(m+1)\epsilon} \mathbb{E}_{\hat{X}}[h(\hat{X}+1) - h(\hat{X})] - 2\mathbb{E}_{\hat{Y}}[h(\hat{Y}+1) - h(\hat{Y})] \tag{B.121}$$

where $\hat{X} \sim \text{Binomial}(K-1, 1-e^{\epsilon}(1-p))$ and $\hat{Y} \sim \text{Binomial}(K-1, p)$. The above

implies

$$\nabla_q f(q;\gamma) \geq 0 \iff e^{(m+1)\epsilon}\mathbb{E}_{\hat{X}}[h(\hat{X}+1)-h(\hat{X})] \geq \mathbb{E}_{\hat{Y}}[h(\hat{Y}+1)-h(\hat{Y})] \quad \text{(B.122)}$$

If $\nabla_q f(q;\gamma) \geq 0$, then since $q \in [0, \frac{1}{1+e^\epsilon}]$, we know that the local maximum given any γ is $(q^*_{local}, q'^*_{local}) = \text{argmax}_{(q,q'):q'=e^\epsilon q, q\in[0,\frac{1}{1+e^\epsilon}]} f(q,q';\gamma) = (\frac{1}{1+e^\epsilon}, \frac{1}{1+e^{-\epsilon}})$. That is,

$$(p^*_{local}, p'^*_{local}) = \underset{(p,p'):1-p'=e^\epsilon(1-p), p\in[\frac{1}{1+e^{-\epsilon}},1]}{\text{argmax}} f(p,p';\gamma) = (1-q^*_{local}, 1-q'^*_{local}) = (\frac{1}{1+e^{-\epsilon}}, \frac{1}{1+e^\epsilon})$$

Notice by Eq. B.117, the above $(\frac{1}{1+e^{-\epsilon}}, \frac{1}{1+e^\epsilon})$ is the local minimum on the first boundary $p = e^\epsilon p'$, $\forall p \in [0, \frac{1}{1+e^{-\epsilon}}]$.

Therefore, given an arbitrary γ function, if it satisfies both of the following:

1. On the boundary $p = e^\epsilon p', \forall p \in [0, \frac{1}{1+e^{-\epsilon}}]$, $\nabla_{p'} f(p';\gamma) \leq 0$
2. On the boundary $1-p' = e^\epsilon(1-p), \forall p \in [\frac{1}{1+e^{-\epsilon}}, 1]$, $\nabla_{q'} f(q';\gamma) \geq 0$ where $q' = 1-p'$

then the global worst case probabilities given this γ is $(p^*, p'^*) = \text{argmax}_{(p,p')\in\mathcal{F}} f(p,p';\gamma) = (0,0)$. Furthermore, since by Eq. B.116, $f(0,0;\gamma) \leq e^{m\epsilon} - 1$ for any γ, this implies DaRRM_γ is $m\epsilon$-differentially private by Lemma 3.3.4.

Now, if γ belongs to the "symmetric form family", by Eq. B.115 and Eq. B.122, the sufficient conditions for γ that enables DaRRM_γ to be $m\epsilon$-differentially private are hence

$$e^\epsilon \mathbb{E}_Y[h(Y+1)-h(Y)] \leq e^{m\epsilon}\mathbb{E}_X[h(X+1)-h(X)], \quad \forall p' \in [0, \frac{1}{1+e^\epsilon}]$$
$$\text{and} \quad e^{(m+1)\epsilon}\mathbb{E}_{\hat{X}}[h(\hat{X}+1)-h(\hat{X})] \geq \mathbb{E}_{\hat{Y}}[h(\hat{Y}+1)-h(\hat{Y})], \quad \forall p \in [\frac{1}{1+e^{-\epsilon}}, 1]$$

where $X \sim \text{Binomial}(K-1, p')$, $Y \sim \text{Binomial}(K-1, e^\epsilon p')$, $\hat{X} \sim \text{Binomial}(K-1, 1-e^\epsilon(1-p))$ and $\hat{Y} \sim \text{Binomial}(K-1, p)$.

□

Lemma B.2.8 (Binomial Expectation Recurrence Relationship (Theorem 2.1 of [122]))**.** *Let $X_{(K-1)} \sim$ Binomial$(K-1, p)$ and $X_{(K)} \sim$ Binomial(K, p). Let $g(x)$ be a function*

with $-\infty < \mathbb{E}[g(X_{(K-1)})] < \infty$ *and* $-\infty < g(-1) < \infty$*, then*

$$Kp\mathbb{E}_{X_{(K-1)}}[g(X_{(K-1)})] = \mathbb{E}_{X_{(K)}}[X_{(K)}g(X_{(K)}-1)] \tag{B.123}$$

Lemma B.2.9. *Given* $i, m, K \in \mathbb{Z}$*,* $K \geq 1$*,* $0 \leq i \leq m \leq K$*, let* $X_{(K)} \sim$ *Binomial*(K, p) *for some* $p \in [0,1]$*, there is*

$$\frac{1}{\binom{K}{m}}\mathbb{E}_{X_{(K)}}\left[\binom{X}{i}\binom{K-X}{m-i}\right] = \binom{m}{i}p^i(1-p)^{m-i} \tag{B.124}$$

Proof of Lemma B.2.9. We show the above statement in Eq. B.124 by induction on K and m.

Base Case: $K = 1$.

1. If $m = 0$, then $i = 0$. $\frac{1}{\binom{1}{0}}\mathbb{E}_{X_{(1)}}[\binom{X}{0}\binom{1-X}{0}] = \mathbb{E}_{X_{(1)}}[1] = 1$, and $\binom{0}{0}p^0(1-p)^0 = 1$.
2. If $m = 1$,
   (a) $i = 0$, $\frac{1}{\binom{1}{1}}\mathbb{E}_{X_{(1)}}[\binom{X}{0}\binom{1-X}{1}] = \mathbb{E}_{X_{(1)}}[1-X] = 1-p$, and $\binom{1}{0}p^0(1-p)^1 = 1-p$
   (b) $i = 1$, $\frac{1}{\binom{1}{1}}\mathbb{E}_{X_{(1)}}[\binom{X}{1}\binom{1-X}{0}] = \mathbb{E}_{X_{(1)}}[X] = p$, and $\binom{1}{1}p^1(1-p)^0 = p$.

Hence, Eq. B.124 holds for the base case.

Induction Hypothesis: Suppose the statement holds for some $K \geq 1$ and $0 \leq i \leq m \leq K$. Consider $1 \leq i \leq m \leq K+1$,

$$\frac{1}{\binom{K+1}{m}}\mathbb{E}_{X_{(K+1)}}\left[\binom{X}{i}\binom{K+1-X}{m-i}\right] \tag{B.125}$$

$$= \frac{1}{\binom{K+1}{m}}\mathbb{E}_{X_{(K+1)}}[\frac{X!}{i!(X-i)!}\frac{(K+1-X)!}{(m-i)!(K+1-X-(m-i))!}] \tag{B.126}$$

$$= \frac{1}{\binom{K+1}{m}i!(m-i)!}\mathbb{E}_{X_{(K+1)}}[X\frac{(X-1)!}{((X-1)-(i-1))!}\frac{(K-(X-1))!}{(K-(X-1)-((m-1)-(i-1)))!}] \tag{B.127}$$

$$= \frac{1}{\binom{K+1}{m}i!(m-i)!}\mathbb{E}_{X_{(K)}}[\frac{X!}{(X-(i-1))!}\frac{(K-X)!}{(K-X-((m-1)-(i-1)))!}] \tag{B.128}$$

(By Lemma B.2.8)

$$= \frac{(i-1)!(m-i)!}{\binom{K+1}{m}i!(m-i)!}\mathbb{E}_{X_{(K)}}[\binom{X}{i-1}\binom{K-X}{(m-1)-(i-1)}] \tag{B.129}$$

$$= \frac{(i-1)!}{\binom{K+1}{m} i!}(K+1)p\binom{K}{m-1}\binom{m-1}{i-1}p^{i-1}(1-p)^{m-i} \tag{B.130}$$

(By Induction Hypothesis)

$$= \frac{m!(K+1-m)!}{(K+1)!i}\frac{K!}{(m-1)!(K-m+1)!}\frac{(m-1)!}{(i-1)!(m-i)!}(K+1)p^i(1-p)^{m-i} \tag{B.131}$$

$$= \frac{m!}{i!(m-i)!}p^i(1-p)^{m-i} = \binom{m}{i}p^i(1-p)^{m-i} \tag{B.132}$$

Now we consider the edge cases when $0 = i \leq m$.

If $i = 0$ and $m = 0$,

$$\frac{1}{\binom{K+1}{0}}\mathbb{E}_{X_{(K+1)}}[\binom{X}{0}\binom{K+1-X}{0}] = 1 \cdot \mathbb{E}_{X_{(K+1)}}[1] = 1 = \binom{0}{0}p^0(1-p)^0 \tag{B.133}$$

If $i = 0$ and $m > 0$,

$$\frac{1}{\binom{K+1}{m}}\mathbb{E}_{X_{(K+1)}}[\binom{K+1-X}{m}] \tag{B.134}$$

$$= \frac{1}{\binom{K+1}{m}}\sum_{x=0}^{K+1}\binom{K+1-x}{m}\binom{K+1}{x}p^x(1-p)^{K+1-x} \tag{B.135}$$

$$= \frac{1}{\binom{K+1}{m}}\sum_{x=0}^{K+1}\binom{K+1-x}{m}\Big(\binom{K}{x} + \binom{K}{x-1}\mathbb{I}\{x \geq 1\}\Big)p^x(1-p)^{K+1-x} \tag{B.136}$$

$$= \frac{1}{\binom{K+1}{m}}\sum_{x=0}^{K}\binom{K+1-x}{m}\binom{K}{x}p^x(1-p)^{K+1-x} + \frac{1}{\binom{K+1}{m}}\sum_{x=1}^{K+1}\binom{K+1-x}{m}\binom{K}{x-1}p^x(1-p)^{K+1-x} \tag{B.137}$$

(Since when $x = K+1$ and $m > 0$, $\binom{K+1-x}{m} = 0$)

$$= \frac{1}{\binom{K+1}{m}}\Big(\sum_{x=0}^{K}\binom{K-x}{m}\binom{K}{x}p^x(1-p)^{K+1-x} + \sum_{x=0}^{K}\binom{K-x}{m-1}\binom{K}{x}p^x(1-p)^{K+1-x}\Big) \tag{B.138}$$

$$+ \frac{1}{\binom{K+1}{m}}\sum_{x=0}^{K}\binom{K-x}{m}\binom{K}{x}p^{x+1}(1-p)^{K-x}$$

$$\text{(Since } \binom{K+1-x}{m} = \binom{K-x}{m} + \binom{K-x}{m-1}\text{)}$$

$$= \frac{1}{\binom{K+1}{m}}\Big((1-p)\mathbb{E}_{X_{(K)}}[\binom{K-X}{m}] + (1-p)\mathbb{E}_{X_{(k)}}[\binom{K-X}{m-1}]\Big) + \frac{1}{\binom{K+1}{m}}p\mathbb{E}_{X_{(K)}}[\binom{K-X}{m}] \tag{B.139}$$

$$= \frac{1}{\binom{K+1}{m}}\Big(\mathbb{E}_{X_{(K)}}[\binom{K-X}{m}] + (1-p)\mathbb{E}_{X_{(K)}}[\binom{K-X}{m-1}]\Big) \tag{B.140}$$

$$= \frac{1}{\binom{K+1}{m}}\Big(\binom{K}{m}(1-p)^m + (1-p)\binom{K}{m-1}(1-p)^{m-1}\Big) \tag{B.141}$$

(By Induction Hypothesis) (B.142)

$$= \frac{1}{\binom{K+1}{m}}\binom{K+1}{m}(1-p)^m \tag{B.143}$$

$$= (1-p)^m \tag{B.144}$$

Hence, Eq. B.124 holds for all $K \geq 1$ and $0 \leq i \leq m \leq K$.

□

Main Result: Privacy Amplification Under a Small m

Lemma B.2.10 (Privacy amplification, $m \leq \frac{K-1}{2}$)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1, with i.i.d. mechanisms $\{M_i\}_{i=1}^K$, $p_i = p$, $p_i' = p'$, $\forall i \in [K]$, the privacy allowance $1 \leq m \leq \frac{K-1}{2}, m \in \mathbb{Z}$ and $\delta = \Delta = 0$. Let the noise function be that*

$$\gamma_{DSub}(l) = \begin{cases} 1 - 2h(l) & \forall l \in \{0, 1, \ldots, \frac{K-1}{2}\} \\ 2h(l) - 1 & \forall l \in \{\frac{K+1}{2}, \ldots, K\} \end{cases} \tag{B.145}$$

where $h : \{0, 1, \ldots, K\} \to [0, 1]$ and $h(l) = \sum_{i=m}^{2m-1} \frac{\binom{l}{i}\binom{K-l}{2m-1-i}}{\binom{K}{2m-1}}$, $\forall l \in \{0, 1, \ldots, K\}$, then Algorithm $\mathsf{DaRRM}_{\gamma_{DSub}}$ is $m\epsilon$-differentially private.

Proof of Lemma B.2.10. First, note γ_{DSub} belongs to the "symmetric form family". We show γ_{DSub} satisfies the two sufficient conditions in Lemma B.2.7 and hence by Lemma B.2.7, $\mathsf{DaRRM}_{\gamma_{DSub}}$ is $m\epsilon$-differentially private. Specifically, we consider $h(l) = \sum_{i=m}^{2m-1} \frac{\binom{l}{i}\binom{K-l}{2m-1-i}}{\binom{K}{2m-1}}$, $\forall l \in \{0, 1, \ldots, K\}$ and $1 \leq m \leq K$.

Two show the first condition is satisfied, let $X_{(K-1)} \sim \text{Binomial}(K-1, p)$ and $Y_{(K-1)} \sim \text{Binomial}(K-1, e^\epsilon p)$, and consider $p \in [0, \frac{1}{1+e^\epsilon}]$.

$$\mathbb{E}_{X_{(K-1)}}[h(X+1)] \tag{B.146}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \mathbb{E}_{X_{(K-1)}}[\binom{X+1}{i}\binom{K-X-1}{2m-1-i}]$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \mathbb{E}_{X_{(K-1)}}[\binom{X}{i}\binom{K-X-1}{2m-1-i} + \binom{X}{i-1}\binom{K-X-1}{2m-1-i}] \tag{B.147}$$

(Since $\binom{X+1}{i} = \binom{X}{i} + \binom{X}{i-1}\mathbb{I}\{i \geq 1\}$)

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\mathbb{E}_{X_{(K-1)}}[\binom{X}{i}\binom{K-1-X}{2m-1-i}] + \mathbb{E}_{X_{(K-1)}}[\binom{X}{i-1}\binom{K-1-X}{(2m-2)-(i-1)}]\right) \tag{B.148}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\binom{K-1}{2m-1}\binom{2m-1}{i}p^i(1-p)^{2m-1-i} + \binom{K-1}{2m-2}\binom{2m-2}{i-1}p^{i-1}(1-p)^{2m-1-i}\right) \tag{B.149}$$

(By Lemma B.2.9)

$$\mathbb{E}_{X_{(K-1)}}[h(X)] \tag{B.150}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \mathbb{E}_{X_{(K-1)}}[\binom{X}{i}\binom{K-X}{2m-1-i}]$$

(Since $\binom{K-X}{2m-1-i} = \binom{K-1-X}{2m-1-i} + \binom{K-1-X}{2m-2-i}$)

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\mathbb{E}_{X_{(K-1)}}[\binom{X}{i}\binom{K-1-X}{2m-1-i}] + \mathbb{E}_{X_{(K-1)}}[\binom{X}{i}\binom{K-1-X}{2m-2-i}]\mathbb{I}\{i \leq 2m-2\}\right) \tag{B.151}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\binom{K-1}{2m-1} \binom{2m-1}{i} p^i (1-p)^{2m-1-i} + \binom{K-1}{2m-2} \binom{2m-2}{i} p^i (1-p)^{2m-2-i} \mathbb{I}\{i \leq \right. \tag{B.152}$$

(By Lemma B.2.9)

Hence, following Eq. B.152 and Eq. B.149,

$$\mathbb{E}_{X_{(K-1)}}[h(X+1) - h(X)] \tag{B.153}$$

$$= \frac{1}{\binom{K}{2m-1}} \Big(\sum_{i=m}^{2m-1} \binom{K-1}{2m-2} \binom{2m-2}{i-1} p^{i-1} (1-p)^{2m-1-i} - \sum_{i=m}^{2m-2} \binom{K-1}{2m-2} \binom{2m-2}{i} p^i (1-p)^{2m-2-} \tag{B.154}$$

$$= \frac{1}{\binom{K}{2m-1}} \Big(\sum_{i=m-1}^{2m-2} \binom{K-1}{2m-2} \binom{2m-2}{i} p^i (1-p)^{2m-2-i} - \sum_{i=m}^{2m-2} \binom{K-1}{2m-2} \binom{2m-2}{i} p^i (1-p)^{2m-2-i} \tag{B.155}$$

$$= \frac{2m-1}{K} \binom{2m-2}{m-1} p^{m-1} (1-p)^{m-1} \tag{B.156}$$

Similarly,

$$\mathbb{E}_{Y_{(K-1)}}[h(Y+1) - h(Y)] = \frac{2m-1}{K} \binom{2m-2}{m-1} (e^\epsilon p)^{m-1} (1-e^\epsilon p)^{m-1} \tag{B.157}$$

Since $p \in [0, \frac{1}{1+e^\epsilon}]$, there is $p(1-p) \geq e^{-\epsilon} e^\epsilon p(1-e^\epsilon p)$. Hence,

$$e^{(m-1)\epsilon} \mathbb{E}_{X_{(K-1)}}[h(X+1) - h(X)] = \frac{2m-1}{K} \binom{2m-2}{m-1} e^{(m-1)\epsilon} p^{m-1} (1-p)^{m-1} \tag{B.158}$$

$$\geq \frac{2m-1}{K} \binom{2m-2}{m-1} e^{(m-1)\epsilon} (e^{-\epsilon} e^\epsilon p(1-e^\epsilon p))^{m-1} \tag{B.159}$$

$$= \frac{2m-1}{K} \binom{2m-2}{m-1} (e^\epsilon p)^{m-1} (1-e^\epsilon p)^{m-1} \tag{B.160}$$

$$= \mathbb{E}_{Y_{(K-1)}}[h(Y+1) - h(Y)] \tag{B.161}$$

implying

$$e^{m\epsilon}\mathbb{E}_{X_{(K-1)}}[h(X+1)-h(X)] \geq e^{\epsilon}\mathbb{E}_{Y_{(K-1)}}[h(Y+1)-h(Y)] \tag{B.162}$$

and the first condition is satisfied.

To show the second condition is satisfied, let $\hat{X}_{(K-1)} \sim \text{Binom}(K-1, 1-e^{\epsilon}(1-p))$ and $\hat{Y}_{(K-1)} \sim \text{Binom}(K-1, p)$, and consider $p \in [\frac{1}{1+e^{-\epsilon}}, 1]$.

$$\mathbb{E}_{\hat{X}_{(K-1)}}[h(\hat{X}+1)] = \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\mathbb{E}_{\hat{X}_{(K-1)}}[\binom{\hat{X}}{i}\binom{K-1-\hat{X}}{2m-1-i}] + \mathbb{E}_{\hat{X}_{(K-1)}}[\binom{\hat{X}}{i-1}\binom{K-1-\hat{X}}{(2m-2)-(i-1)} \tag{B.163}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\binom{K-1}{2m-1}\binom{2m-1}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-1-i}\right. \tag{B.164}$$

$$\left. + \binom{K-1}{2m-2}\binom{2m-2}{i-1}(1-e^{\epsilon}(1-p))^{i-1}(e^{\epsilon}(1-p))^{2m-1-i}\right)$$

By Lemma B.2.9

and

$$\mathbb{E}_{\hat{X}_{(K-1)}}[h(\hat{X})] = \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\mathbb{E}_{\hat{X}_{(K-1)}}[\binom{\hat{X}}{i}\binom{K-1-\hat{X}}{2m-1-i}] + \mathbb{E}_{\hat{X}_{(K-1)}}[\binom{\hat{X}}{i}\binom{K-1-\hat{X}}{2m-2-i}]\mathbb{I}\{i \leq 2m- \tag{B.165}$$

$$= \frac{1}{\binom{K}{2m-1}} \sum_{i=m}^{2m-1} \left(\binom{K-1}{2m-1}\binom{2m-1}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-1-i}\right. \tag{B.166}$$

$$\left. + \binom{K-1}{2m-2}\binom{2m-2}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-2-i}\mathbb{I}\{i \leq 2m-2\}\right)$$

By Lemma B.2.9

Hence, following Eq. B.164 and Eq. B.166,

$$\mathbb{E}_{\hat{X}_{(K-1)}}[h(\hat{X}+1)-h(\hat{X})] \tag{B.167}$$

$$= \frac{1}{\binom{K}{2m-1}}\Big(\sum_{i=m}^{2m-1}\binom{K-1}{2m-2}\binom{2m-2}{i-1}(1-e^{\epsilon}(1-p))^{i-1}(e^{\epsilon}(1-p))^{2m-1-i} \tag{B.168}$$

$$- \sum_{i=m}^{2m-2}\binom{K-1}{2m-2}\binom{2m-2}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-2-i}\Big)$$

$$= \frac{1}{\binom{K}{2m-1}}\Big(\sum_{i=m-1}^{2m-2}\binom{K-1}{2m-2}\binom{2m-2}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-2-i} \tag{B.169}$$

$$- \sum_{i=m}^{2m-2}\binom{K-1}{2m-2}\binom{2m-2}{i}(1-e^{\epsilon}(1-p))^{i}(e^{\epsilon}(1-p))^{2m-2-i}\Big)$$

$$= \frac{2m-1}{K}\binom{2m-2}{m-1}(1-e^{\epsilon}(1-p))^{m-1}(e^{\epsilon}(1-p))^{m-1} \tag{B.170}$$

Similarly,

$$\mathbb{E}_{\hat{Y}_{(K-1)}}[h(\hat{Y}+1)-h(\hat{Y})] = \frac{2m-1}{K}\binom{2m-2}{m-1}p^{m-1}(1-p)^{m-1} \tag{B.171}$$

Hence,

$$e^{(m+1)\epsilon}\mathbb{E}_{\hat{X}_{(K-1)}}[h(\hat{X}+1)-h(\hat{X})] = e^{(m+1)\epsilon}\frac{2m-1}{K}\binom{2m-2}{m-1}(1-e^{\epsilon}(1-p))^{m-1}(e^{\epsilon}(1-p))^{m-1} \tag{B.172}$$

$$\geq \frac{2m-1}{K}\binom{2m-2}{m-1}(1-e^{\epsilon}(1-p))^{m-1}e^{(m-1)\epsilon}(1-p)^{m-1} \tag{B.173}$$

$$= \frac{2m-1}{K}\binom{2m-2}{m-1}(e^{\epsilon}-e^{2\epsilon}(1-p))^{m-1}(1-p)^{m-1} \tag{B.174}$$

Note that

$$e^{\epsilon}-e^{2\epsilon}(1-p) = e^{\epsilon}-e^{2\epsilon}+e^{2\epsilon}p \geq p \tag{B.175}$$

$$\iff (e^{\epsilon}+1)(e^{\epsilon}-1)p \geq e^{\epsilon}(e^{\epsilon}-1) \tag{B.176}$$

$$\iff p \geq \frac{e^{\epsilon}}{e^{\epsilon}+1} = \frac{1}{1+e^{-\epsilon}} \tag{B.177}$$

and the condition needs to hold for $p \in [\frac{1}{1+e^{-\epsilon}}, 1]$.

Therefore, following Eq. B.174,

$$e^{(m+1)\epsilon}\mathbb{E}_{\hat{X}_{(K-1)}}[h(\hat{X}+1) - h(\hat{X})] \geq \frac{2m-1}{K}\binom{2m-2}{m-1}p^{m-1}(1-p)^{m-1} \tag{B.178}$$

$$= \mathbb{E}_{\hat{Y}_{(K-1)}}[h(\hat{Y}+1) - h(\hat{Y})] \tag{B.179}$$

implying the second condition is satisfied.

Therefore, by Lemma B.2.7, $\mathsf{DaRRM}_{\gamma_{DSub}}$ is $m\epsilon$-differentially private.

□

B.2.3 Comparing the Utility of Subsampling Approaches

Intuitively, if we subsample $2m-1$ mechanisms, the utility is higher than that of the naïve subsampling approach which outputs the majority based on only m mechanisms. To complete the story, we formally compare the utility of outputting the majority of $2m-1$ subsampled mechanisms (Theorem 3.4.1) and outputting the majority of m subsampled mechanisms (simple composition, Theorem 3.2.2) in the i.i.d. mechanisms and pure differential privacy setting, fixing the output privacy loss to be $m\epsilon$.

Lemma B.2.11. *Consider Problem 3.1.1 with i.i.d. mechanisms* $\{M_i\}_{i=1}^K$, *i.e.,* $p = p_i = \Pr[M_i(\mathcal{D}) = 1], p' = p'_i = \Pr[M_i(\mathcal{D}') = 1], \forall i \in [K]$. *Let* $\gamma_1 : \{0, 1, \ldots, K\} \to [0,1], \gamma_2 : \{0, 1, \ldots, K\} \to [0,1]$ *be two functions that are both symmetric around* $\frac{K}{2}$. *If* $1 \geq \gamma_1(l) \geq \gamma_2(l) \geq 0, \forall l \in \{0, \ldots, K\}$, *then* $\mathcal{E}(\mathsf{DaRRM}_{\gamma_1}) \leq \mathcal{E}(\mathsf{DaRRM}_{\gamma_2})$.

Proof. Recall $\mathcal{S} = \{S_1, \ldots, S_K\}$, where $S_i \sim M_i(\mathcal{D})$, is the set of observed outcomes from the mechanisms $\{M_i\}_{i=1}^K$. By Definition 3.2.4, for any γ that is symmetric around $\frac{K}{2}$, the error of DaRRM_γ is

$$\mathcal{E}(\mathsf{DaRRM}_\gamma) = \Big|\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D}) = 1] - \Pr[g(\mathcal{S}) = 1]\Big| \tag{B.180}$$

$$= \Big|\sum_{l=\frac{K+1}{2}}^{K}\Big(\gamma(l) + \frac{1}{2}(1-\gamma(l))\Big)\cdot\alpha_l + \sum_{l=0}^{\frac{K-1}{2}}\frac{1}{2}(1-\gamma(l))\cdot\alpha_l - \sum_{l=\frac{K+1}{2}}^{K}\alpha_l\Big| \tag{B.181}$$

$$= \Big| \sum_{l=\frac{K+1}{2}}^{K} \Big(\frac{1}{2}\gamma(l) - \frac{1}{2}\Big) \cdot \alpha_l + \sum_{l=0}^{\frac{K-1}{2}} \Big(\frac{1}{2} - \frac{1}{2}\gamma(l)\Big) \cdot \alpha_l \Big| \tag{B.182}$$

$$= \Big| \frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} (1 - \gamma(l)) \cdot (\alpha_l - \alpha_{K-l}) \Big| \tag{B.183}$$

where $\alpha_l = \binom{K}{l} p^l (1-p)^{K-l}$, $\forall l \in \{0, 1, \ldots, K\}$ and recall $p = \Pr[M_i(\mathcal{D}) = 1]$, $\forall i \in [K]$.

For any $l \geq \frac{K+1}{2}$,

1. If $p = 0$ or $p = 1$, $\alpha_l = \alpha_{K-l}$.
2. Otherwise, for $p \in (0, 1)$,
   (a) If $p \geq \frac{1}{2}$,

$$\frac{\alpha_l}{\alpha_{K-l}} = \frac{p^l (1-p)^{K-l}}{p^{K-l}(1-p)^l} = p^{2l-K}(1-p)^{K-2l} = (\underbrace{\frac{p}{1-p}}_{\geq 1})^{\underbrace{2l-K}_{\geq 0}} \geq 1, \quad \Rightarrow \alpha_l \geq \alpha_{K-l} \tag{B.184}$$

   (b) If $p < \frac{1}{2}$,

$$\frac{\alpha_l}{\alpha_{K-l}} = (\underbrace{\frac{p}{1-p}}_{\leq 1})^{\underbrace{2l-K}_{\geq 0}} \leq 1, \quad \Rightarrow \alpha_l \leq \alpha_{K-l} \tag{B.185}$$

Hence, if $p \geq \frac{1}{2}$, then $\alpha_l \geq \alpha_{K-l}, \forall l \geq \frac{K+1}{2}$. Since $\gamma_1(l) \geq \gamma_2(l), \forall l \in \{0, \ldots, K\}$, $1 - \gamma_1(l) \leq 1 - \gamma_2(l)$, and so

$$\mathcal{E}(\mathsf{DaRRM}_{\gamma_1}) = \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1 - \gamma_1(l)) \cdot (\alpha_l - \alpha_{K-l}) \leq \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1 - \gamma_2(l)) \cdot (\alpha_l - \alpha_{K-l}) = \mathcal{E}(\mathsf{DaRRM}_{\gamma_2}) \tag{B.186}$$

Similarly, if $p < \frac{1}{2}$, then $\alpha_l \leq \alpha_{K-l}, \forall l \geq \frac{K+1}{2}$ and

$$\mathcal{E}(\mathsf{DaRRM}_{\gamma_1}) = \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1-\gamma_1(l)) \cdot (\alpha_{K-l} - \alpha_l) \leq \sum_{l=\frac{K+1}{2}}^{K} \frac{1}{2}(1-\gamma_2(l)) \cdot (\alpha_{K-l} - \alpha_l) = \mathcal{E}(\mathsf{DaRRM}_{\gamma_2})) \tag{B.187}$$

Therefore,

$$\mathcal{E}(\mathsf{DaRRM}_{\gamma_1}) \leq \mathcal{E}(\mathsf{DaRRM}_{\gamma_2}) \tag{B.188}$$

□

Since $\gamma_{DSub}(l) \geq \gamma_{Sub}(l)$, $\forall l \in \{0, 1, \ldots, K\}$, by Lemma B.2.11, $\mathcal{E}(\mathsf{DaRRM}_{\gamma_{DSub}}) \leq \mathcal{E}(\mathsf{DaRRM}_{\gamma_{Sub}})$ — that is, outputting $2m-1$ mechanisms has a higher utility than outputting m mechanisms.

B.3 Details of Section 3.5: Optimizing the Noise Function γ in DaRRM

B.3.1 Deriving the Optimization Objective

For any γ function that is symmetric around $\frac{K}{2}$, we can write the optimization objective as

$$\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}[\mathcal{E}(\mathsf{DaRRM}_\gamma)] \tag{B.189}$$

$$= \mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}[|\Pr[\mathsf{DaRRM}_\gamma(\mathcal{D})=1]-\Pr[g(\mathcal{S})=1]|] \tag{B.190}$$

$$= \mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\Big|\sum_{l=\frac{K+1}{2}}^{K}\Big(\alpha_l\cdot(\gamma(l)+\frac{1}{2}(1-\gamma(l)))-\alpha_l\Big)+\sum_{l=0}^{\frac{K-1}{2}}\alpha_l\cdot\frac{1}{2}(1-\gamma(l))\Big|\right] \tag{B.191}$$

$$= \mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\Big|\sum_{l=0}^{\frac{K-1}{2}}\alpha_l(\frac{1}{2}\gamma(l)-\frac{1}{2})+\sum_{l=\frac{K+1}{2}}^{K}\alpha_l(\frac{1}{2}-\frac{1}{2}\gamma(l))\Big|\right] \tag{B.192}$$

The above follows by conditioning on $\mathcal{L}=l\in\{0,1,\ldots,K\}$, i.e. the sum of observed outcomes in

$$= \mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\Big|\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l-\alpha_{K-l})\,(1-\gamma(l))\Big|\right] \tag{B.193}$$

The above follows by symmetry of γ

Furthermore, notice the objective is symmetric around 0, and can be written as

$$\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l-\alpha_{K-l})\,(1-\gamma(l))\right] \tag{B.194}$$

$$= \frac{1}{2}\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\sum_{l=\frac{K+1}{2}}^{K}\Big((\alpha_l-\alpha_{K-l})-(\alpha_l-\alpha_{K-l})\gamma(l)\Big)\right] \tag{B.195}$$

$$= \underbrace{\frac{1}{2}\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - \alpha_{K-l})\right]}_{:=A} \underbrace{-\frac{1}{2}\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - \alpha_{K-l})\gamma(l)\right]}_{:=B} \tag{B.196}$$

Since expression A in Eq. B.196 does not involve γ, we only need to optimize expression B in Eq. B.196. That is,

$$-\frac{1}{2}\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[\sum_{l=\frac{K+1}{2}}^{K}(\alpha_l - \alpha_{K-l})\gamma(l)\right] \tag{B.197}$$

$$= -\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}\mathbb{E}_{p_1,p_2,\ldots,p_K\sim\mathcal{T}}\left[(\alpha_l - \alpha_{K-l})\right]\cdot\gamma(l) \tag{B.198}$$

Eq. B.198 is the optimization objective we use in the experiments. We see the optimization objective is linear in γ.

Note in the general setting, $\mathcal{L}(\mathcal{D}) \sim \text{PoissonBinomial}(p_1, p_2, \ldots, p_K)$, where recall $\mathcal{L}(\mathcal{D})$ is the sum of observed outcomes on dataset $\mathcal{D}$, and hence, $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ is the pmf of the Poisson Binomial distribution at $l \in \{0, 1, \ldots, K\}$.

B.3.2 Practical Approximation of the Objective

Since the optimization objective in Eq. B.197 requires taking an expectation over $p_1, \ldots, p_K$, and this invovles integrating over K variables, which can be slow in practice, we propose the following approximation to efficiently compute the objective. We start with a simple idea to compute the objective, by sampling p_i's from $[0, 1]$ and take an empirical average of the objective value over all subsampled sets of $p_1, \ldots, p_K$ as the approximation of the expectation in Section B.3.2. However, we found this approach is less numerically stable. We then propose the second approach to approximate the objective in Section B.3.2, which approximates the integration over p_i's using the rectangular rule instead of directly approximating the objective value. We use the second approximation approach in our experiments and empirically demonstrates its effectiveness. **Note approximating the optimization objective does not affect the privacy guarantee.**

Approximation via Direct Sampling of p_i's

One straightforward way of efficiently computing an approximation to the optimization objective is as follows:

Algorithm 5 Straightforward Approximation of the Optimization Objective

Input: # mechanisms $K \in \mathbb{N}$, # iterations $T \in \mathbb{N}$, noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$
for $t = 1, 2, \ldots, T$ **do**
  Sample $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_K \sim \mathcal{T}$
  $\widehat{\mathcal{L}} \leftarrow \text{PoissonBinomail}(\hat{p}_1, \ldots, \hat{p}_K)$
  $\hat{\alpha}_l \leftarrow \Pr[\widehat{\mathcal{L}} = l], \forall l \{0, \ldots, K\}$
  $g_t \leftarrow -\frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} (\hat{\alpha}_l - \hat{\alpha}_{K-l}) \cdot \gamma(l)$
end for
Return $\frac{1}{T} \sum_{t=1}^{T} g_t$

However, we found this approximation is not very numerically stable even for $T = 10000$ in the experiments and so we propose to adopt the second approximation as follows.

Approximating the Integration Over p_i's

Consider the following surrogate objective:

$$-\frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} \int_{0.5}^{1} \int_{0.5}^{1} \cdots \int_{0.5}^{1} (\alpha_l - \alpha_{K-l}) dp_1 dp_2 \ldots dp_K \cdot \gamma(l) \tag{B.199}$$

where we approximate the integration instead of directly approximating the objective value. The approximation of the integration is based on the rectangular rule and that the Poisson Binomial distribution is invariant to the order of its probability parameters.

First, we discretize the integration over p_i's: pick $\tau = 50$ points representing probabilities between $[0.5, 1)$ with equal distance $\theta = \frac{0.5}{\tau}$. Denote this set of points as $\mathcal{W}$. We pick only $\tau = 50$ samples to ensure the distance between each sample, i.e., θ, is not too small; or this can cause numerical instability. For each $l \in$

$\{\frac{K+1}{2}, \frac{K+1}{2}+1, \ldots, K\}$, we want to compute an approximated coefficient for $\gamma(l)$ as follows:

$$\int_{0.5}^{1}\int_{0.5}^{1}\cdots\int_{0.5}^{1}(\alpha_l - \alpha_{K-l})dp_1dp_2\ldots dp_K \approx \sum_{p_1\in\mathcal{W}}\sum_{p_2\in\mathcal{W}}\cdots\sum_{p_K\in\mathcal{W}}(\alpha_l - \alpha_{K-l}) \quad \text{(B.200)}$$

which approximates integration over a K-dimensional grid $\mathcal{W}^K$.

The idea is then to sample points from this K-dimensional grid $\mathcal{W}^K$ and compute an empirical mean of the integration based on the sample probabilities for $p_1, \ldots, p_K$ from $\mathcal{W}^K$ as the approximation of the integration in the objective.

Let $(s_1, s_2, \ldots, s_K)$ be randomly sampled probability values from $\mathcal{W}^K$ and we want to compute $(\alpha_l - \alpha_{K-l})$ for all l based on $(p_1, \ldots, p_K) = (s_1, \ldots, s_K)$. To apply the rectangular rule, since the grid of probabilities is K-dimensional, the weight of $(\alpha_l - \alpha_{K-l})$ in the approximate integration is θ^K. Furthermore, observe that α_l is the pmf at l from a Poison Binomial distribution in our case, and PoissonBinomial$(p_1, \ldots, p_K) \overset{dist.}{\sim}$ PoissonBinomial$(\pi(p_1, \ldots, p_K))$, where π denotes a permutation of $p_1, \ldots, p_K$ and $\overset{dist.}{\sim}$ denotes "the same distribution". Hence, with a single probability sample $(s_1, \ldots, s_K)$, we can indeed compute $\alpha_l - \alpha_{K-l}$ for each l at $K!$ points from the grid $\mathcal{W}^K$, since they all have the same value. Therefore, we should set the weight of $\alpha_l - \alpha_{K-l}$ in the approximate integration as $w = \theta^K \cdot K!$. Furthermore, since the order of $(p_1, \ldots, p_K)$ does not affect the objective value, there is a total of (τ choose K with replacement) $= \binom{\tau+K-1}{K} := P$ different points in the grid $\mathcal{W}^K$.

In summary, the integration based approximation of the objective proceeds as follows:

Algorithm 6 Integration Based Approximation of the Optimization Objective

Input: # mechanisms $K \in \mathbb{N}$, # iterations $T = 10000 \in \mathbb{N}$, noise function $\gamma : \{0, 1, \ldots, K\} \to [0, 1]$, $\tau = 50$: # samples between $[0.5, 1)$ to form the set $\mathcal{W}$
$\theta \leftarrow 0.5/\tau$ distance between samples
$w \leftarrow \theta^K \cdot K!$
$P \leftarrow \binom{\tau+K-1}{K}$
for $t = 1, 2, \ldots, T$ **do**
  Sample probabilities $(s_1, s_2, \ldots, s_K) \sim \mathcal{W}^K$
  $\widehat{\mathcal{L}} \sim \text{PoissonBinomial}(s_1, s_2, \ldots, s_K)$
  $\hat{\alpha}_l \leftarrow \Pr[\widehat{\mathcal{L}} = l], \forall l \in \{0, 1, \ldots, K\}$
  $g_t \leftarrow -\frac{1}{2} \sum_{l=\frac{K+1}{2}}^{K} w \cdot (\hat{\alpha}_l - \hat{\alpha}_{K-l}) \cdot \gamma(l)$
end for
Return $\frac{P}{N} \sum_{t=1}^{T} g_t$

B.3.3 Reducing # Constraints from ∞ to a Polynomial Set

Lemma B.3.1 (Restatement of Lemma 3.5.1)**.** *Consider using DaRRM (Algorithm 1) to solve Problem 3.1.1 and let f be the privacy cost objective as defined in Lemma 3.3.4. Given an arbitrary noise function γ, let the worst case probabilities be*

$$(p_1^*, \ldots, p_K^*, p_1'^*, \ldots, p_K'^*) = \underset{\{(p_i, p_i')\}_{i=1}^K}{\text{argmax}} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$$

Then, each pair $(p_i^*, p_i'^*), \forall i \in [K]$ satisfies

$$(p_i^*, p_i'^*) \in \{(0,0), (1,1), (0,\Delta), (\Delta,0), (1-\Delta,1), (1,1-\Delta), (\frac{e^\epsilon+\Delta}{e^\epsilon+1}, \frac{1-\Delta}{e^\epsilon+1}), (\frac{1-\Delta}{e^\epsilon+1}, \frac{e^\epsilon+\Delta}{e^\epsilon+1})\}$$

Furthermore, when $\delta > 0$, there exists a finite vector set $\mathcal{P}$ of size $O(K^7)$ such that if $\beta = \max_{\{(p_i, p_i')\}_{i=1}^K \in \mathcal{P}} f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$, then $f(p_1^*, \ldots, p_K^*, p_1'^*, \ldots, p_K'^*; \gamma) \leq \beta$. When $\delta = 0$, the size of $\mathcal{P}$ can be reduced to $O(K^3)$.

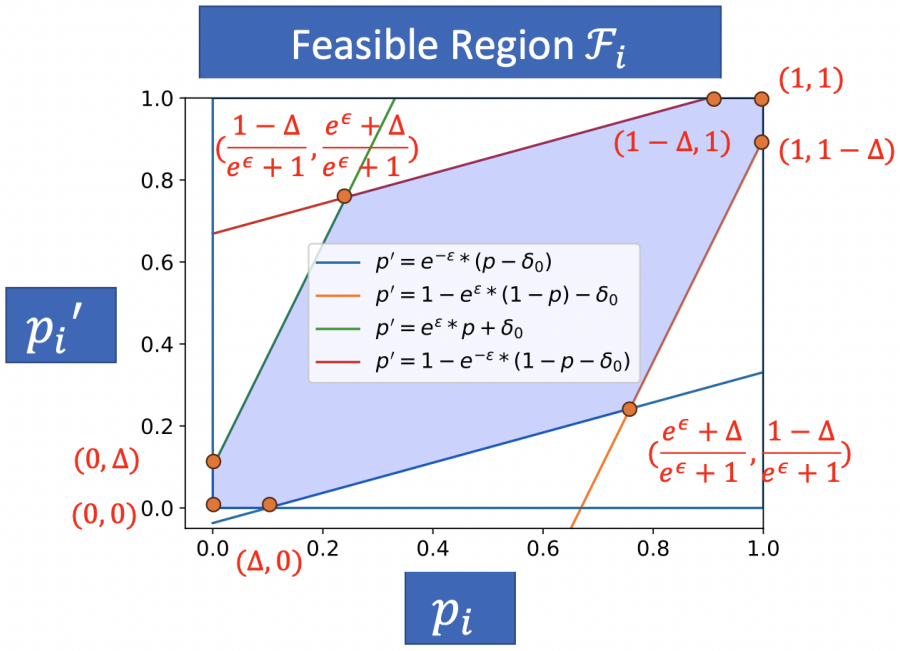

Figure B.3: An illustration of the feasible region $\mathcal{F}_i$.

Proof. **Part I: Reducing # privacy constraints from ∞ to exponentially many.**

Consider (p_i, p_i') for an arbitrary $i \in [K]$ and fixing $(p_j, p_j'), \forall j \neq i$. Given any noise function γ, recall the privacy cost objective $f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma)$ (see Lemma 3.3.4), is

$$f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma) = \sum_{l=0}^{\frac{K-1}{2}} (e^{m\epsilon}\alpha_l' - \alpha_l) \cdot \gamma(l) + \sum_{l=\frac{K+1}{2}}^{K} (\alpha_l - e^{m\epsilon}\alpha_l') \cdot \gamma(l)$$

and the privacy constraints are of the form

$$f(p_1, \ldots, p_K, p_1', \ldots, p_K'; \gamma) \leq e^{m\epsilon} - 1 + 2\delta$$

where recall that $\alpha_l = \Pr[\mathcal{L}(\mathcal{D}) = l]$ is a function of $\{p_i\}_{i=1}^K$ and $\alpha_l' = \Pr[\mathcal{L}(\mathcal{D}') = l]$ is a function of $\{p_i'\}_{i=1}^K$, $\forall l \in \{0, 1, \ldots, K\}$ and $\mathcal{L}(\mathcal{D})$, $\mathcal{L}(\mathcal{D}')$ are the sum of observed outcomes on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. By Lemma 3.3.4, γ needs

to make the above privacy constraint hold for all possible $\{(p_i, p'_i)\}_{i=1}^K$ to make DaRRM_γ $(m\epsilon, \delta)$-differentially private. This is equivalent to saying, γ needs to ensure $\max_{\{(p_i,p'_i\}_{i=1}^K} f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma) \leq e^{m\epsilon} - 1 + 2\delta$.

Notice that the sum of observed outcomes follows a Poisson Binomial distribution, i.e., $\mathcal{L}(\mathcal{D}) \sim \text{PoissonBinomial}(p_1, \ldots, p_K)$ and $\mathcal{L}(\mathcal{D}') \sim \text{PoissonBinomial}(p'_1, \ldots, p'_K)$. Hence, by the pmf of the Poisson Binomial distribution[1], the privacy cost objective f is linear in each p_i and p'_i, fixing all (p_j, p'_j), $\forall j \neq i$. Since each mechanism M_i is (ϵ, Δ)-differentially private, by definition, (p_i, p'_i) satisfies all of the following:

$$p_i \leq e^\epsilon p'_i + \Delta, \quad p'_i \leq e^\epsilon p + \Delta$$
$$1 - p_i \leq e^\epsilon (1 - p'_i) + \Delta, \quad 1 - p'_i \leq e^\epsilon (1 - p_i) + \Delta$$

That is, (p_i, p'_i) lies in a feasible region $\mathcal{F}_i$ (see Figure B.3). Note the constraints on (p_i, p'_i), that is, the boundaries of $\mathcal{F}_i$, are linear in p_i and p'_i. And so the optimization problem $(p_i^*, p_i'^*) = \text{argmax}_{(p_i,p'_i)} f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma)$, which finds the worst case probabilities in (p_i, p'_i), is a Linear Programming (LP) problem in (p_i, p'_i) for $i \in [K]$. This implies $(p_i^*, p_i'^*)$ has to be on one of the eight corners of $\mathcal{F}_i$ — that is $(p_i^*, p_i'^*) \in \{(0,0), (1,1), (0,\Delta), (\Delta,0), (1-\Delta,1), (1,1-\Delta), (\frac{e^\epsilon+\Delta}{e^\epsilon+1}, \frac{1-\Delta}{e^\epsilon+1}), (\frac{1-\Delta}{e^\epsilon+1}, \frac{e^\epsilon+\Delta}{e^\epsilon+1})\} := \mathcal{C}$. Since all (p_i, p'_i) and (p_j, p'_j), for $i \neq j$, are independent, we can search for the worst case probabilities by searching for $(p_i^*, p_i'^*) \in \mathcal{C}$, instead of searching for $(p_i, p'_i) \in \mathcal{F}_i, \forall i \in [K]$. Therefore, the infinitely many privacy constraints are now reduced to only 8^K to optimize for the best γ function that maximizes the utility of DaRRM_γ, while ensuring the output is $m\epsilon$-differentially private.

Part II: Reducing # privacy constraints from exponentially many to a polynomial set.

To further reduce the number of privacy constraints in optimization, observe that the Poisson Binomial distribution is invariant under the permutation of its parameters. That is, $\text{PoissonBinomial}(p_1, \ldots, p_K) \overset{dist.}{\sim} \text{PoissonBinomial}(\pi(p_1, \ldots, p_K))$, for some permutation π and $\overset{dist.}{\sim}$ means "follows the same distribution". Similarly, $\text{PoissonBinomial}(p'_1, \ldots, p'_K) \overset{dist.}{\sim} \text{PoissonBinomial}(\pi(p'_1, \ldots, p'_K))$.

The above observation implies if we have one privacy constraint $f(p_1 = v_1, \ldots, p_K =$

[1]See, e.g. https://en.wikipedia.org/wiki/Poisson_binomial_distribution, for the pmf of Poisson Binomial distribution.

$v_K, p'_1 = v'_1, \ldots, p'_K = v'_K; \gamma) \leq e^{m\epsilon} - 1 + 2\delta$, for some $\{(v_i, v'_i)\}_{i=1}^K \in \mathcal{C}^K$, then any privacy constraint $f(p_1 = s_1, \ldots, p_K = s_K, p'_1 = s'_1, \ldots, p'_K = s'_K; \gamma) \leq e^{m\epsilon} - 1 + 2\delta$, where $(s_1, \ldots, s_K) = \pi_1(v_1, \ldots, v_K)$, $(s'_1, \ldots, s'_K) = \pi(v'_1, \ldots, v'_K)$, for permutations π_1 and π_2, is redundant.

Therefore, there is a vector set $\mathcal{P}$, where each probability vector $(p_1, \ldots, p_K, p'_1, \ldots, p'_K)$ in $\mathcal{P}$ is constructed by setting $(p_1, p'_1), (p_2, p'_2), \ldots, (p_K, p'_K) = (v_1, v_2, \ldots, v_K)$, where $v_i \in \mathcal{C}, \forall i \in [K]$, such that vectors constructed by $(p_1, p'_1), (p_2, p'_2), \ldots, (p_K, p'_K) = \pi(v_1, v_2, \ldots, v_K)$ is not in $\mathcal{P}$. Note $|\mathcal{P}|$ = (8 chooses K with replacement) = $\binom{K+8-1}{K} = O(K^7)$. If we can restrict our search for the worst case probabilities to this set $\mathcal{P}$ — that is, solving for $\beta := \max_{\{(p_i, p'_i)\}_{i=1}^K \in \mathcal{P}} f(p_1, \ldots, p_K, p'_1, \ldots, p'_K; \gamma)$, then $f(p^*_1, \ldots, p^*_K, p'^*_1, \ldots, p'^*_K; \gamma) \leq \beta$. This implies we only need $O(K^7)$ privacy constraints to optimize for the best noise function γ in DaRRM, while making sure DaRRM_γ is $m\epsilon$-differentially private.

Note if $\Delta = 0$, i.e., the mechanism M_i's are pure differentially private, the feasible region $\mathcal{F}_i$ in which (p_i, p'_i) lies has only 4 corners instead of 8. This implies $(p^*_i, p'^*_i) \in \mathcal{C} = \{(0,0), (1,1), (\frac{e^\epsilon}{e^\epsilon+1}, \frac{1}{e^\epsilon+1}), (\frac{1}{e^\epsilon+1}, \frac{e^\epsilon}{e^\epsilon+1})\}$. Hence, in this case, $|\mathcal{P}|$ = (4 choose K with replacement) = $\binom{K+4-1}{K} = O(K^3)$, which implies we only need $O(K^3)$ privacy constraints to optimize for the best noise function γ in DaRRM.

□

B.4 Full Experiment Results

B.4.1 Optimized γ in Simulations

Comparison Using General Composition

The general composition (Theorem 3.2.3) indicates less total privacy loss than simple composition (Theorem 3.2.2) when the number of folds, m, is large, or when the failure probability δ is large. To enable meaningful comparison against general composition, we consider a larger K and a larger failure probability δ.

Consider $K = 35, \epsilon = 0.1, \Delta = 10^{-5}$. By general composition, if one outputs the majority of M subsampled mechanisms for some $M < K$, the majority output is $(\epsilon_{opt}, \delta_{opt})$-differentially private, where

$$\epsilon_{opt} = \min\left\{M\epsilon, \frac{(e^\epsilon - 1)\epsilon M}{e^\epsilon + 1} + \epsilon\sqrt{2M\log(e + \frac{\sqrt{M\epsilon^2}}{\delta'})}, \frac{(e^\epsilon - 1)\epsilon M}{e^\epsilon + 1} + \epsilon\sqrt{2M\log(\frac{1}{\delta'})}\right\}$$
$$\delta_{opt} = 1 - (1-\delta)^M(1-\delta')$$

for some $\delta' \geq 0$. We set this as the privacy guarantee of all majority ensembling algorithms. That is, if we want the majority output to be $(m\epsilon, \delta)$-differentially private, we set

$$m = \frac{\epsilon_{opt}}{\epsilon} = \min\left\{M, \frac{(e^\epsilon - 1)M}{e^\epsilon + 1} + \sqrt{2M\log(e + \frac{\sqrt{M\epsilon^2}}{\delta'})}, \frac{(e^\epsilon - 1)M}{e^\epsilon + 1} + \sqrt{2M\log(\frac{1}{\delta'})}\right\}$$

and $\delta = 1 - (1-\delta)^M(1-\delta')$ accordingly. The parameters τ and λ to compute p_{const} in RR (see Section B.1.1) are set to be

$$\tau = \min\left\{K, \frac{(e^\epsilon - 1)K}{e^\epsilon + 1} + \sqrt{2K\log(e + \frac{\sqrt{K\epsilon^2}}{\delta'})}, \frac{(e^\epsilon - 1)K}{e^\epsilon + 1} + \sqrt{2K\log(\frac{1}{\delta'})}\right\}$$

and $\lambda = 1 - (1-\delta)^K(1-\delta')$.

In the experiments, we consider $M = \{10, 13, 15, 20\}$ and $\delta' = 0.1$; and γ_{opt} is computed using a uniform prior $\mathcal{T}$.

All values of the parameters of the private ensembling algorithms we use in the

experiment are listed in the table:

# Subsampled mechanisms	M	10	13	15	20
Privacy allowance	m	6.4521	7.5742	8.2708	9.8823
Parameter of constant γ	τ	14.0328	14.0328	14.0328	14.0328
Parameter of constant γ	λ	0.1003	0.1003	0.1003	0.1003
Overall privacy loss	$m\epsilon$	0.6452	0.7574	0.8271	0.9882
Overall failure probability	δ	0.1001	0.1001	0.1001	0.1002

Table B.1: All parameter values. Note that all the private ensembling algorithms we compare in the experiment is required to be $(m\epsilon, \delta)$-differentially private. Here, $K = 35, \epsilon = 0.1, \Delta = 10^{-5}$ and $\delta' = 0.1$.

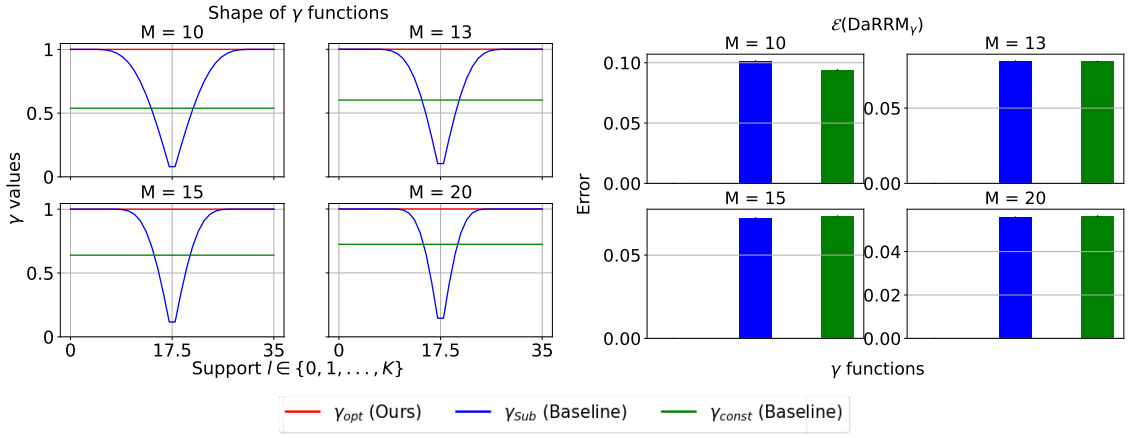

Figure B.4: Plots of the shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: the optimized γ_{Sub}, and the baselines γ_{Sub} (corresponding to subsampling) and γ_{const} (corresponding to RR). Here, $K = 35, M \in \{10, 13, 15, 20\}$, $\Delta = 10^{-5}$, $\epsilon = 0.1$, $\delta' = 0.1$.

Comparison in Pure Differential Privacy Settings

Consider the pure differential privacy setting, where $\Delta = \delta = 0$. Note in this setting, it is known that simple composition is tight.

To compute an optimized γ_{opt} in DaRRM, since we have shown the number of constraints is $O(K^3)$ if $\Delta = \delta = 0$ (see Lemma 3.5.1), we can set K to be larger. Here, we present results for $K \in \{11, 101\}$ and $\epsilon = 0.1$.

Again, we compare the shape of different γ and the corresponding $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ under those γ functions, fixing the total privacy loss to be $m\epsilon$. γ_{opt} is computed using a uniform prior $\mathcal{T}$.

Since the subsampling mechanism from Section 3.4 with privacy amplification applies to this setting, we compare four different γ noise functions here:

1. γ_{opt} (Ours): optimized γ function using our optimization framework
2. γ_{Sub} (Baseline): the γ function that corresponds to outputting the majority of m out K subsampled mechanisms
3. γ_{DSub} (Baseline): the γ function that corresponds to outputting $2m-1$ subsampled mechanisms from Theorem 3.4.1, aka., Double Subsampling (DSub)
4. γ_{const} (Baseline): the constant γ function that corresponds to the classical Randomized Response (RR) algorithm

Setting 1. $K = 11$, $m \in \{1, 3, 5, 7, 9, 11\}$.

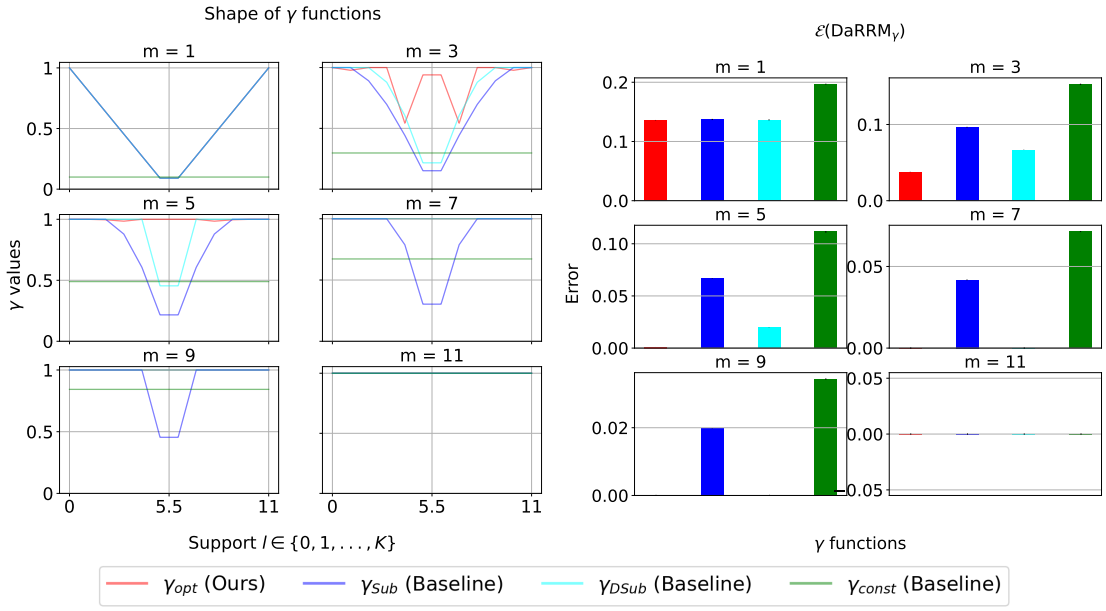

Figure B.5: Plots of shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: the optimized γ_{Opt}, the baselines γ_{Sub} and γ_{DSub} (Theorem 3.4.1), and the constant γ_{const} (corresponding to RR). Here, $K = 11, m \in \{1, 3, 5, 7, 9, 11\}$, $\epsilon = 0.1$ and $\delta = \Delta = 0$. Note when $m \in \{7, 9\}$, the cyan line (γ_{DSub}) and the red line (γ_{opt}) overlap. When $m = 11$, all lines overlap. Observe that when $m \geq \frac{K+1}{2}$, that is, $m \in \{7, 9, 11\}$ in this case, the above plots suggest both γ_{opt} and γ_{DSub} achieve the minimum error at 0. This is consistent with our theory.

Setting 2. $K = 101, m \in \{10, 20, 30, 40, 60, 80\}$.

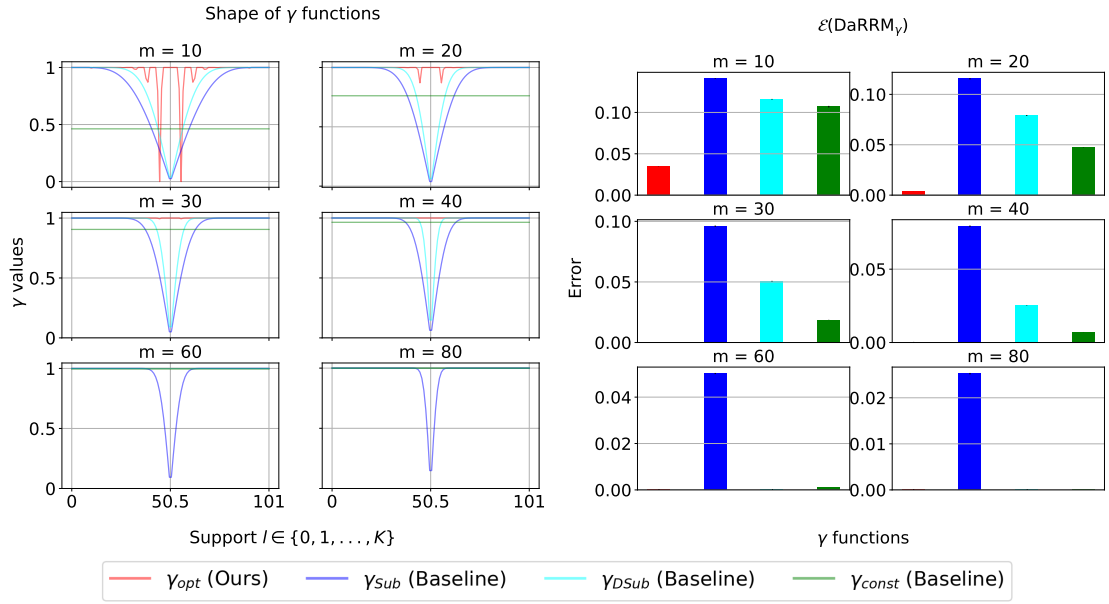

Figure B.6: Plots of shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: the optimized γ_{Opt}, the baselines γ_{Sub} and γ_{DSub} (Theorem 3.4.1), and the constant γ_{const} (corresponding to RR). Here, $K = 101, m \in \{10, 20, 30, 40, 60, 80\}$, $\epsilon = 0.1$ and $\delta = \Delta = 0$.

Comparison Using Different Prior Distributions

When optimizing γ that maximizes the utility in DaRRM, recall that the objective takes an expectation over p_i's for $p_i \sim \mathcal{T}$, where $\mathcal{T}$ is some distribution and $p_i = \Pr[M_i(\mathcal{D}) = 1]$. The previous experiments assume we do not have access to any prior knowledge about p_i's and hence $\mathcal{T}$ is the uniform distribution, i.e., Uniform($[0, 1]$). However, when one has knowledge about the mechanisms, one can set a proper prior $\mathcal{T}$ to further maximize the utility of DaRRM.

In this section, let $\mathcal{T}_U$ denote Uniform($[0, 1]$) and we present results considering a different prior distribution, which we call $\mathcal{T}_P$, as follows. Suppose our prior belief is that each mechanism M_i has a clear tendency towards voting 0 or 1, i.e., p_i is far from 0.5. Let $\mathcal{T}_P$ be Uniform($[0, 0.3] \cup [0.7, 1]$).

To optimize γ under $\mathcal{T}_P$, we change the approximate optimization objective in Eq. B.199, which optimizes γ under $\mathcal{T}_U$, to be the following,

$$-\frac{1}{2}\sum_{l=\frac{K+1}{2}}^{K}\int_{0.7}^{1}\int_{0.7}^{1}\cdots\int_{0.7}^{1}(\alpha_l - \alpha_{K-l})dp_1dp_2\ldots dp_K \cdot \gamma(l) \tag{B.201}$$

Setting. $K = 11, m \in \{3, 5\}$, $\epsilon = 0.1$, $\delta = \Delta = 0$.

We compare the shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions:

1. γ_{opt-U} denote the γ function optimized under $p_i \sim \mathcal{T}_U$
2. γ_{opt-P} denote the γ function optimized under $p_i \sim \mathcal{T}_P$
3. γ_{Sub}, corresponding to the subsampling baseline
4. γ_{const}, corresponding to the RR baseline

Note when we compute the error, we take the expectation w.r.t. the actual p_i distributions, regardless of the prior used to optimize γ. In the experiments, we consider three different actual p_i distributions:"

1. "Actual: Uniform($[0, 1]$)": $p_i \sim \mathcal{T}_U, \forall i \in [K]$
2. "Actual: $p_i = 0.5$": $p_i = 0.5, \forall i \in [K]$

   This setting implies the mechanisms do not have a clear majority
3. "Actual: Uniform($[0, 0.1]$)": $p_i \sim \text{Uniform}([0, 0.1]), \forall i \in [K]$

   This setting implies the mechanisms have a clear majority (i.e., 0)

Since our prior $\mathcal{T}_P$ is closer to Uniform($[0, 0.1]$) (i.e., there is a clear majority), we would expect $\mathcal{E}(\mathsf{DaRRM}_{\gamma_{opt-P}})$ to be the lowest when $p_i \sim \text{Uniform}[0, 0.1]$, but to be higher than $\mathcal{E}(\mathsf{DaRRM}_{\gamma_{opt-U}})$ when $p_i \sim \text{Uniform}([0, 1])$ or $p_i = 0.5$. The results are presented in Figure B.7.

B.4.2 Private Semi-Supervised Knowledge Transfer

More Details about the Baseline GNMax [87]

The GNMax aggregation mechanism for majority ensembling of *non-private* teachers proceeds as follows (Section 4.1 of [87]): on input x,

$$M_\sigma(x) = \underset{i}{\text{argmax}}\{n_i(x) + \mathcal{N}(0, \sigma^2)\} \tag{B.202}$$

where $n_i(x)$ is # teachers who vote for class i.

How to set σ in GNMax?

Section 4.1 of [87] states the GNMax mechanism is $(\lambda, \lambda/\sigma^2)$-Renyi differentially private (RDP), for all $\lambda \geq 1$. RDP bounds can be converted to DP bounds as follows:

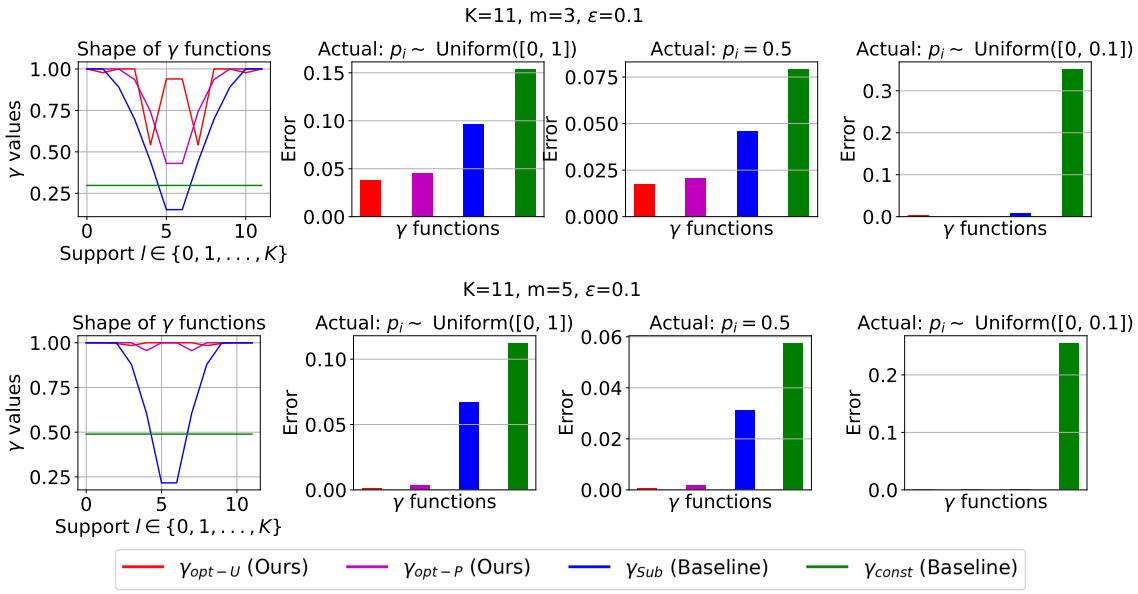

Figure B.7: Comparison of the shape and $\mathcal{E}(\mathsf{DaRRM}_\gamma)$ of different γ functions: 1) γ optimized under prior $\mathcal{T}_U$, 2) γ optimized under prior $\mathcal{T}_P$, 3) γ_{Sub} (corresponding to the subsampling baseline) and 4) γ_{const} (corresponding to the RR baseline). Here, $K = 11, m \in \{3, 5\}, \epsilon = 0.1$. Observe that if the prior $\mathcal{T}_P$ used in optimizing γ is closer to the actual distribution of p_i's, there is additional utility gain (i.e., decreased error); otherwise, we slightly suffer a utility loss (i.e., increased error), compared to optimize γ under the $\mathcal{T}_U$ prior. Furthermore, regardless of the choice of the prior distribution $\mathcal{T}$ in optimizing γ, DaRRM_γ with an optimized γ achieves a lower error compared to the the baselines.

Theorem B.4.1 (RDP to DP (Theorem 5 of [87])). *If a mechanism M guarantees (λ, ϵ)-RDP, then M guarantees $(\epsilon + \frac{\log 1/\delta}{\lambda-1}, \delta)$-differential privacy for $\delta \in (0, 1)$.*

Therefore, GNMax with parameter σ^2 guarantees $(\frac{\lambda}{\sigma^2} + \frac{\log 1/\delta}{\lambda-1}, \delta)$-differential privacy, $\forall \lambda \geq 1$. Given m, ϵ, Δ, we want to choose λ and σ^2 here so that the output of GNMax is $(m\epsilon, m\Delta)$-differentially private. Here, $\delta = m\Delta$.

We first obtain a valid range of λ. Since $m\epsilon \geq 0$, $\frac{\lambda}{\sigma^2} + \frac{\log 1/\delta}{\lambda-1} \geq 0$ and so $\lambda \geq \frac{\log 1/\delta}{m\epsilon} + 1 := \lambda_{min}$. And $\sigma^2 = \frac{\lambda}{m\epsilon - \frac{\log 1/\delta}{\lambda-1}}$. Since the smaller σ^2 is, the higher the utility, we perform a grid search over $\lambda \in [\lambda_{min}, 500]$, with discretized λ values of equal distance 0.5, to find the minimum σ^2_{min}. For the $(m\epsilon, m\Delta)$ values used in the experiments, we observe σ^2 decreases first and then increases as λ increases, as shown in Figure B.8. The λ and σ_{min} values in the RDP bound of Gaussian noise to compute the privacy loss of GNMax's output we use in the experiments are presented in Table B.2.

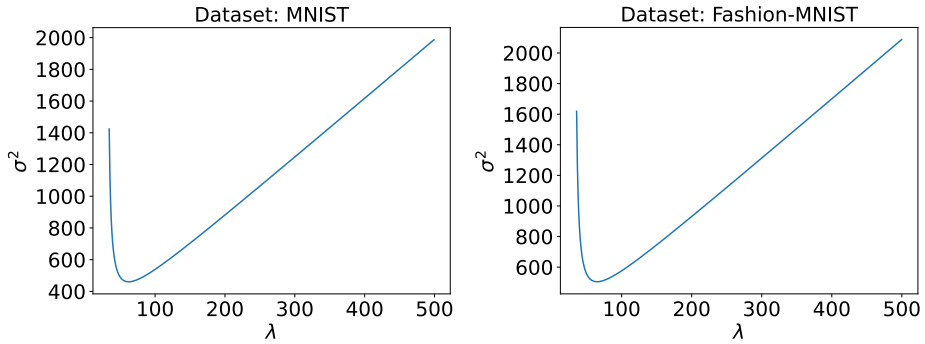

Figure B.8: Plots of λ vs. σ^2 in the Gaussian RDP privacy bound. The goal is to choose a λ value that minimizes σ^2. It is not hard to see the value of σ^2 decreases at first and then increases as λ increases.

	Privacy Loss Per Query $(m\epsilon, m\Delta)$	λ	σ_{min}
MNIST	$(0.2676, 0.0003)$	34.31	21.46
Fashion-MNIST	$(0.2556, 0.0003)$	35.74	22.46

Table B.2: Parameters of the RDP bound of Gaussian noise to compute the privacy loss of GNMax's output.

A Note on the Data-dependent Privacy Loss Bound

Algorithm 7 Compute Tighter Privacy Loss

1: Input: Std. of Gaussian noise σ, Privacy parameter λ, # teachers K, # classes C, # votes per class $\{n_i\}_{i=1}^{C}$
2: $\mathcal{B} \leftarrow \{\}$ bound candidates
3: **for** $i = 1, 2, \ldots, K$ **do**
4: $\quad q^{(i)} \leftarrow \frac{1}{2}\sum_{i \neq i^*} \text{erfc}(\frac{n_{i^*} - n_i}{2\sigma})$
5: $\quad \mu_2^{(i)} \leftarrow \sigma \cdot \sqrt{\log 1/q^{(i)}}$, $\mu_1^{(i)} \leftarrow \mu_2^{(i)} + 1$
6: $\quad \epsilon_1^{(i)} \leftarrow \frac{\mu_1^{(i)}}{\sigma^2}$, $\epsilon_2^{(i)} \leftarrow \frac{\mu_2^{(i)}}{\sigma^2}$
7: $\quad q_{ub}^{(i)} \leftarrow \exp((\mu_2^{(i)} - 1)\epsilon_2^{(i)})/(\frac{\mu_1^{(i)}}{\mu_1^{(i)}-1} \cdot \frac{\mu_2^{(i)}}{\mu_2^{(i)}-1})^{\mu_2^{(i)}}$
8: $\quad$ **if** $q^{(i)} < 1$ and $\mu_1^{(i)} \geq \lambda$ and $\mu_2 > 1$ and $q^{(i)} \leq q_{ub}^{(i)}$ **then**
9: $\quad\quad A^{(i)} \leftarrow (1 - q^{(i)})/(1 - q^{(i)} \cdot \exp(\epsilon_2^{(i)})^{\frac{\mu_2^{(i)}-1}{\mu_2^{(i)}}})$
10: $\quad\quad B^{(i)} \leftarrow \exp(\epsilon_1^{(i)})/(q^{(i)})^{\frac{1}{\mu_1^{(i)}-1}}$
11: $\quad\quad \text{DataDependentBound} \leftarrow \frac{1}{\lambda-1} \cdot \left((1 - q^{(i)}) \cdot (A^{(i)})^{\lambda-1} + q^{(i)} \cdot (B^{(i)})^{\lambda-1}\right)$
12: $\quad\quad \mathcal{B} \leftarrow \mathcal{B} \cup \text{DataDependentBound}$
13: $\quad$ **else**
14: $\quad\quad \text{GaussianBound} \leftarrow \frac{\lambda}{\sigma^2}$
15: $\quad\quad \mathcal{B} \leftarrow \mathcal{B} \cup \text{GaussianBound}$
16: $\quad$ **end if**
17: **end for**
18: Return $\min \mathcal{B}$

[87] gives a potentially tighter data-dependent bound on the privacy loss using GNMax to output the majority of non-private teacherss votes. We give a clean pseudo-code on computing the data-dependent privacy loss bound in Algorithm 7, based on the lemmas and theorems in [87]. Given privacy parameters σ, λ and the teacher votes per class $\{n_i\}_{i=1}^{C}$ for C classes, the data-dependent bound can be empirically evaluated and compared against the Gaussian privacy loss bound. The smaller one is the final privacy loss. We empirically find that the condition of the data-dependent bound (line 8 in Algorithm 7) is not satisfied when K and the number of classes C are small, e.g., $K = 11, C = 2$ as in our case, even if all teachers agree on the same output. And so in the experiments, we can only apply the Gaussian privacy loss bound (line 14).

Additional Results for Private Semi-Supervised Knowledge Transfer

$m = 1$.

Dataset	# Queries	Privacy loss per query $(\epsilon_{query}, \delta_{query})$	Total privacy loss over Q queries $(\epsilon_{total}, \delta_{total})$
MNIST	$Q = 20$	$(0.0892, 0.0001)$	$(1.704, 0.002)$
	$Q = 50$		$(2.837, 0.005)$
	$Q = 100$		$(4.202, 0.010)$
Fashion MNIST	$Q = 20$	$(0.0852, 0.0001)$	$(1.620, 0.002)$
	$Q = 50$		$(2.695, 0.005)$
	$Q = 100$		$(3.988, 0.010)$

Table B.3: The privacy loss per query to the teachers and the total privacy loss over Q queries. Note the total privacy loss is computed by general composition, where we set $\delta' = 0.0001$.

Dataset	MNIST			Dataset	Fashion-MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)	# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.54 (0.11)	0.68 (0.07)	**0.74 (0.08)**	$Q = 20$	0.56 (0.10)	**0.92 (0.05)**	0.89 (0.06)
$Q = 50$	0.51 (0.07)	**0.67 (0.05)**	0.66 (0.05)	$Q = 50$	0.52 (0.05)	0.89 (0.04)	**0.92 (0.03)**
$Q = 100$	0.57 (0.03)	**0.71 (0.03)**	0.69 (0.04)	$Q = 100$	0.56 (0.04)	0.89 (0.04)	**0.91 (0.04)**

Table B.4: Accuracy of the predicted labels of Q query samples on datasets MNIST (on the left) and Fashion-MNIST (on the right). We report the mean and one std. in parentheses over 10 random draws of the query samples from the test dataset. Note each prediction on the query sample is $(\epsilon_{total}, \delta_{total})$-differentially private. Note in this case where $m = 1$, by Lemma 3.3.2, subsampling achieves the optimal error/utility. Hence, there is not much difference in terms of accuracy between DaRRM$_{\gamma_{Sub}}$ and DaRRM$_{\gamma_{opt}}$ as expected.

$m = 5$.

Dataset	# Queries	Privacy loss per query $(\epsilon_{query}, \delta_{query})$	Total privacy loss over Q queries $(\epsilon_{total}, \delta_{total})$
MNIST	$Q = 20$	$(0.4460, 0.0005)$	$(8.920, 0.010)$
	$Q = 50$		$(18.428, 0.025)$
	$Q = 100$		$(28.926, 0.049)$
Fashion MNIST	$Q = 20$	$(0.4260, 0.0005)$	$(8.520, 0.010)$
	$Q = 50$		$(17.398, 0.025)$
	$Q = 100$		$(27.223, 0.049)$

Table B.5: The privacy loss per query to the teachers and the total privacy loss over Q queries. Note the total privacy loss is computed by general composition, where we set $\delta' = 0.0001$.

Dataset	MNIST			Dataset	Fashion-MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)	# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.73 (0.11)	0.76 (0.09)	**0.84 (0.07)**	$Q = 20$	0.72 (0.10)	0.96 (0.04)	**0.97 (0.04)**
$Q = 50$	0.75 (0.07)	0.82 (0.04)	**0.83 (0.04)**	$Q = 50$	0.72 (0.08)	0.96 (0.02)	**0.97 (0.02)**
$Q = 100$	0.72 (0.04)	0.79 (0.05)	**0.83 (0.03)**	$Q = 100$	0.72 (0.06)	**0.97 (0.01)**	**0.97 (0.01)**

Table B.6: Accuracy of the predicted labels of Q query samples on datasets MNIST (on the left) and Fashion-MNIST (on the right). We report the mean and one std. in parentheses over 10 random draws of the query samples from the test dataset. Note each prediction on the query sample is $(\epsilon_{total}, \delta_{total})$-differentially private. With the same per query privacy loss (and hence the same total privacy loss over Q samples), DaRRM$_{\gamma_{opt}}$ achieves the highest accuracy compared to the other two baselines.

$m = 7$.

Dataset	# Queries	Privacy loss per query $(\epsilon_{query}, \delta_{query})$	Total privacy loss over Q queries $(\epsilon_{total}, \delta_{total})$
MNIST	$Q = 20$	$(0.6244, 0.0007)$	$(12.488, 0.014)$
	$Q = 50$		$(28.392, 0.035)$
	$Q = 100$		$(45.683, 0.068)$
Fashion MNIST	$Q = 20$	$(0.5964, 0.0007)$	$(11.928, 0.014)$
	$Q = 50$		$(26.738, 0.035)$
	$Q = 100$		$(42.873, 0.068)$

Table B.7: The privacy loss per query to the teachers and the total privacy loss over Q queries. Note the total privacy loss is computed by general composition, where we set $\delta' = 0.0001$.

Dataset	MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.79 (0.07)	0.80 (0.09)	**0.85 (0.08)**
$Q = 50$	0.80 (0.05)	0.82 (0.05)	**0.85 (0.04)**
$Q = 100$	0.80 (0.04)	0.80 (0.04)	**0.83 (0.03)**

Dataset	Fashion-MNIST		
# Queries	GNMax (Baseline)	DaRRM$_{\gamma_{Sub}}$ (Baseline)	DaRRM$_{\gamma_{opt}}$ (Ours)
$Q = 20$	0.79 (0.07)	0.95 (0.04)	**0.96 (0.04)**
$Q = 50$	0.79 (0.05)	0.96 (0.03)	**0.97 (0.03)**
$Q = 100$	0.79 (0.03)	**0.96 (0.02)**	**0.96 (0.02)**

Table B.8: Accuracy of the predicted labels of Q query samples on datasets MNIST (on the left) and Fashion-MNIST (on the right). We report the mean and one std. in parentheses over 10 random draws of the query samples from the test dataset. Note each prediction on the query sample is $(\epsilon_{total}, \delta_{total})$-differentially private. With the same per query privacy loss (and hence the same total privacy loss over Q samples), DaRRM$_{\gamma_{opt}}$ achieves the highest accuracy compared to the other two baselines.

Appendix C

Private Incremental Gradient (IG) Methods with Public Data

C.1 All Proof Details

C.1.1 Useful Lemmas

Lemma C.1.1 (Lemma 3.6 of [69]). *Given a convex and differentiable function* $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ *satisfying* $\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ *for some* $L > 0$, *then* $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\frac{\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|^2}{2L} \leq B_g(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

Lemma C.1.2 (Lemma E.1 of [69]). *Under Assumption 4.3.2,*

$$\frac{1}{n}\sum_{i=2}^{n} L_i \Big\| \sum_{j=1}^{i-1} \nabla f_j(\mathbf{x}^*) \Big\|^2 \leq n^2 L \sigma_{any}^2$$

Lemma C.1.3 (Slight Extension of Lemma 6.2 of [69]). *Given a sequence* $d^{(1)}, d^{(2)}, \ldots, d^{(K)}, d^{(K+1)}$,

suppose there exist positive constants a, b, c *and a sequence* $r^{(1)}, r^{(2)}, \ldots, r^{(k)}$ *satisfying*

$$d^{(k+1)} \leq \frac{a}{k} + b(1 + \log k) + c \sum_{l=2}^{k} \frac{d_l}{k - l + 2} + \sum_{l=1}^{k} r^{(l)}, \quad \forall k \in [K] \tag{C.1}$$

then the following inequality holds

$$d^{(k+1)} \leq \Big(\frac{a}{k} + b(1 + \log k)\Big) \sum_{i=0}^{k-1} (2c(1 + \log k))^i + \sum_{l=1}^{k} r^{(l)}, \quad \forall k \in [K] \tag{C.2}$$

Proof. We use induction to show Eq. C.2.

Base Case: for $k = 1$, by Eq. C.1, $d^{(2)} \leq a + b + r^{(1)}$, which also satisfies Eq. C.2.

Induction Hypothesis: suppose Eq. C.2 holds for 1 to $k - 1$ (where $2 \leq k \leq K$), i.e.,

$$d^{(l)} \leq \Big(\frac{a}{l-1} + b(1 + \log(l-1))\Big) \sum_{i=0}^{l-2} (2c(1 + \log(l-1)))^i + \sum_{l=1}^{l-1} r^{(l)}, \quad \forall 2 \leq l \leq k \tag{C.3}$$

which implies

$$d^{(l)} \leq \Big(\frac{a}{l-1} + b(1 + \log k)\Big) \sum_{i=0}^{l-2} (2c(1 + \log k))^i + \sum_{l=1}^{l-1} r^{(l)}, \quad \forall 2 \leq l \leq k \tag{C.4}$$

Now for $d^{(k+1)}$, by Eq. C.1,

$$d^{(k+1)} \leq \frac{a}{k} + b(1 + \log k) + c \sum_{l=2}^{k} \frac{d_l}{k - l + 2} + \sum_{l=1}^{k} r^{(l)} \tag{C.5}$$

$$\leq \frac{a}{c} + ac \sum_{l=2}^{k} \sum_{i=0}^{i-2} \frac{(2c(1 + \log k))^i}{(k - l + 2)(l - 1)} \tag{C.6}$$

$$+ b(1 + \log k)\Big(1 + c \sum_{l=2}^{k} \sum_{i=0}^{l-2} \frac{(2c(1 + \log k))^i}{k - l + 2}\Big) + \sum_{l=1}^{k} r^{(l)}$$

Note that

$$c\sum_{l=2}^{k}\sum_{i=0}^{i-2}\frac{(2c(1+\log k))^i}{(k-l+2)(l-1)} = c\sum_{i=0}^{k-2}(2c(1+\log k))^i\Big(\sum_{l=2+i}^{k}\frac{1}{(k-l+2)(l-1)}\Big) \tag{C.7}$$

$$= \frac{c}{k+1}\sum_{i=0}^{k-2}(2c(1+\log k))^i\Big(\sum_{l=2+i}^{k}\frac{1}{k-l+2}+\frac{1}{l-1}\Big) \tag{C.8}$$

$$\leq \frac{c}{k+1}\sum_{i=0}^{k-2}(2c(1+\log k))^i\sum_{l=1}^{k}\frac{2}{l} \tag{C.9}$$

$$\leq \frac{\sum_{i=0}^{k-2}(2c(1+\log k))^{i+1}}{k+1} \tag{C.10}$$

$$\leq \frac{\sum_{i=0}^{k-1}(2c(1+\log k))^{i}}{k+1} \tag{C.11}$$

$$\leq \frac{\sum_{i=1}^{k-1}(2c(1+\log k))^{i}}{k} \tag{C.12}$$

and

$$c\sum_{l=2}^{k}\sum_{i=0}^{l-2}\frac{(2c(1+\log k))^i}{k-l+2} = c\sum_{i=0}^{k-2}(2c(1+\log k))^i\sum_{l=2+i}^{k}\frac{1}{k-l+2} \leq c\sum_{i=0}^{k-2}(2c(1+\log k))^i\sum_{l=1}^{k}\frac{1}{l} \tag{C.13}$$

$$\leq c(1+\log k)\sum_{i=0}^{k-2}(2c(1+\log k))^i \leq \sum_{i=0}^{k-2}(2c(1+\log k))^{i+1} \tag{C.14}$$

$$\leq \sum_{i=1}^{k-1}(2c(1+\log k))^i \tag{C.15}$$

Combining Eq. C.6, Eq. C.12 and Eq. C.15,

$$d^{(k+1)} \leq \frac{a}{k}+b(1+\log k)(1+\sum_{i=1}^{k-1}(2c(1+\log k))^i)+\sum_{i=1}^{k}r^{(i)} \tag{C.16}$$

$$= \Big(\frac{a}{k}+b(1+\log k)\Big)\sum_{i=0}^{k-1}(2c(1+\log k))^i+\sum_{i=1}^{k}r^{(i)} \tag{C.17}$$

which finishes the induction.

□

C.1.2 One Epoch Convergence

Lemma C.1.4. *Under Assumption 4.3.1, for any $s \in [K]$ and $\mathbf{z} \in \mathbb{R}^d$, Algorithm 2 guarantees*

$$\begin{aligned}\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \leq \mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] \\ + \frac{\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2\right]}{2n\eta} - (\frac{1}{2n\eta} + \frac{\mu_\psi}{2})\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right] - \frac{1}{2n\eta}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)} - \mathbf{x}\right. \\ + \frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)})\right] - \mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)})\right]\right)\end{aligned} \tag{C.18}$$

where the expectation is taken w.r.t. the injected noise $\{\rho_i^{(s)}\}_{i=1}^n$.

Proof of Lemma C.1.4. It suffices to consider only $\mathbf{x} \in \mathrm{dom}(\psi)$.

Let $\mathbf{g}^{(s)} = \sum_{i=1}^n \left(\nabla f_i^{(s)}(\mathbf{x}_i^{(s)}) + \rho_i^{(s)}\right)$. According to the update rule, $\mathbf{x}_{n+1}^{(s)} = \mathbf{x}_1^{(s)} - \eta \cdot \mathbf{g}^{(s)}$. Observe that

$$\begin{aligned}\mathbf{x}_1^{(s+1)} &= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{n\psi(\mathbf{x}) + \frac{\|\mathbf{x} - \mathbf{x}_{n+1}^{(s)}\|^2}{2\eta}\right\} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{n\psi(\mathbf{x}) + \frac{\|\mathbf{x} - \mathbf{x}_1^{(s)} - \eta\cdot\mathbf{g}^{(s)}\|^2}{2\eta}\right\} \\ &= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{n\psi(\mathbf{x}) + \frac{\|\mathbf{x} - \mathbf{x}_1^{(s)}\|^2 + \eta^2\|\mathbf{g}^{(s)}\|^2 + 2\langle\mathbf{x} - \mathbf{x}_1^{(s)}, \eta\mathbf{g}^{(s)}\rangle}{2\eta}\right\} \\ &= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{n\psi(\mathbf{x}) + \frac{\|\mathbf{x} - \mathbf{x}_1^{(s)}\|^2}{2\eta} + \langle\mathbf{x} - \mathbf{x}_1^{(s)}, \mathbf{g}^{(s)}\rangle\right\}\end{aligned}$$

By the first-order optimality condition, there exists $\nabla\psi(\mathbf{x}_1^{(s+1)}) \in \partial\psi(\mathbf{x}_1^{(s+1)})$ such that

$$n\nabla\psi(\mathbf{x}_1^{(s+1)}) + \mathbf{g}^{(s)} + \frac{\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}}{\eta} = \mathbf{0} \iff \mathbf{g}^{(s)} = -n\nabla\psi(\mathbf{x}_1^{(s+1)}) + \frac{\mathbf{x}_1^{(s)} - \mathbf{x}_1^{(s+1)}}{\eta}$$

Therefore, for $\mathbf{z} \in \mathrm{dom}(\psi)$,

$$\langle \mathbf{g}^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle \tag{C.19}$$

$$= n\langle \nabla\psi(\mathbf{x}_1^{(s+1)}), \mathbf{z} - \mathbf{x}_1^{(s+1)} \rangle + \frac{1}{\eta}\langle \mathbf{x}_1^{(s)} - \mathbf{x}_1^{(s+1)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle$$

$$\overset{\text{(a)}}{\leq} n\Big(\psi(\mathbf{z}) - \psi(\mathbf{x}_1^{(s+1)}) - \frac{\mu_\psi}{2}\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\Big) + \frac{1}{\eta}\langle \mathbf{x}_1^{(s)} - \mathbf{x}_1^{(s+1)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle \tag{C.20}$$

$$= n\Big(\psi(\mathbf{z}) - \psi(\mathbf{x}_1^{(s+1)}) - \frac{\mu_\psi}{2}\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\Big) + \frac{1}{2\eta}\Big(\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2 - \|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2 - \|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2\Big) \tag{C.21}$$

$$= n\Big(\psi(\mathbf{z}) - \psi(\mathbf{x}_1^{(s+1)})\Big) + \frac{\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2}{2\eta} - (\frac{1}{2\eta} + \frac{n\mu_\psi}{2})\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2 - \frac{1}{2\eta}\|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2 \tag{C.22}$$

where (a) is by the strong convexity of ψ.

By the definition of $\mathbf{g}^{(s)}$,

$$\langle \mathbf{g}^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle = \langle \sum_{i=1}^{n} \Big(\nabla f_i^{(s)}(\mathbf{x}_i^{(s)}) + \rho_i^{(s)}\Big), \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle \tag{C.23}$$

$$= \sum_{i=1}^{n} \langle \nabla f_i^{(s)}(\mathbf{x}_i^{(s)}), \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle + \sum_{i=1}^{n} \langle \rho_i^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle \tag{C.24}$$

Since

$$B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)}) = f_i^{(s)}(\mathbf{x}_1^{(s+1)}) - f_i^{(s)}(\mathbf{x}_i^{(s)}) - \langle \nabla f_i^{(s)}(\mathbf{x}_i^{(s)}), \mathbf{x}_1^{(s+1)} - \mathbf{x}_i^{(s)} \rangle$$

$$B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)}) = f_i^{(s)}(\mathbf{z}) - f_i^{(s)}(\mathbf{x}_i^{(s)}) - \langle \nabla f_i^{(s)}(\mathbf{x}_i^{(s)}), \mathbf{z} - \mathbf{x}_i^{(s)} \rangle$$

there is

$$\sum_{i=1}^{n} \langle f_i^{(s)}(\mathbf{x}_i^{(s)}), \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle = \sum_{i=1}^{n} \Big(f_i^{(s)}(\mathbf{x}_1^{(s+1)}) - f_i^{(s)}(\mathbf{z}) - B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)}) + B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)})\Big) \tag{C.25}$$

$$= nF^{(s)}(\mathbf{x}_1^{(s+1)}) - nF^{(s)}(\mathbf{z}) - \sum_{i=1}^{n} \Big(B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)}) - B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)})\Big) \tag{C.26}$$

Hence, plugging Eq. C.25 back to Eq. C.24, there is

$$\langle \mathbf{g}^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle \tag{C.27}$$
$$= nF^{(s)}(\mathbf{x}_1^{(s+1)}) - nF^{(s)}(\mathbf{z}) - \sum_{i=1}^{n} \left(B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)}) - B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)}) \right) + \sum_{i=1}^{n} \langle \rho_i^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle$$

Now, by Eq. C.22 and Eq. C.27, after rearranging

$$F^{(s)}(\mathbf{x}_1^{(s+1)}) - F^{(s)}(\mathbf{z}) \leq \frac{\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2}{2n\eta} - (\frac{1}{2n\eta} + \frac{\mu_\psi}{2})\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2 - \frac{1}{2n\eta}\|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2 \tag{C.28}$$
$$+ \frac{1}{n} \sum_{i=1}^{n} \left(B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)}) - B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)}) \right) + \frac{1}{n} \sum_{i=1}^{n} \langle \rho_i^{(s)}, \mathbf{x}_1^{(s+1)} - \mathbf{z} \rangle$$

Taking expectation w.r.t. $\{\rho_i^{(s)}\}_{i=1}^n$ on both sides, and note that $\mathbb{E}[\rho_i^{(s)}] = 0, \forall i \in [n]$,

$$\mathbb{E}\left[F^{(s)}(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F^{(s)}(\mathbf{z})\right] \tag{C.29}$$
$$\leq \frac{\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2\right]}{2n\eta} - (\frac{1}{2n\eta} + \frac{\mu_\psi}{2})\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right] - \frac{1}{2n\eta}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2\right] \tag{C.30}$$
$$+ \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)})\right] - \mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)})\right] \right)$$

Therefore, for any $\mathbf{z} \in \mathbb{R}^d$,

$$\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \tag{C.31}$$
$$= \left(\mathbb{E}\left[F^{(s)}(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F^{(s)}(\mathbf{z})\right] \right) + \left(\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)}) - F^{(s)}(\mathbf{x}_1^{(s+1)})\right] \right) - \left(\mathbb{E}\left[F(\mathbf{z})\right] - \mathbb{E}\left[F^{(s)}(\mathbf{z})\right] \right)$$
$$\leq \mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] \tag{C.32}$$
$$+ \frac{\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2\right]}{2n\eta} - (\frac{1}{2n\eta} + \frac{\mu_\psi}{2})\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right] - \frac{1}{2n\eta}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2\right]$$

$$+\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})\right]-\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\right]\right)$$

□

Lemma C.1.5. *Under Assumption 4.3.1, 4.3.2 and 4.3.7, for any $s\in[K]$ and $\forall \mathbf{z}\in\mathbb{R}^d$, if $\eta\le\frac{1}{n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}}}$, Algorithm 2 guarantees*

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})\right]-\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\right]\right) \\ &\le L^{(s)}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2\right]+5\eta^2\frac{1}{n}\sum_{i=1}^{n-1}L_{i+1}^{(s)}(C_i^{(s)})^2 \\ &\quad+10\eta^2n^2(L^{(s)})^2\mathbb{E}\left[B_F(\mathbf{z},\mathbf{x}^*)\right]+5\eta^2n^2L^{(s)}\sigma_{any}^2+5\eta^2L^{(s)}nd(\sigma^{(s)})^2\end{aligned} \tag{C.33}$$

where the expectation is taken w.r.t. the injected noise and $L^{(s)}\in\{L,\widetilde{L}\}$.

Proof of Lemma C.1.5. By Lemma C.1.1,

$$B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})\le\frac{L_i^{(s)}}{2}\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_i^{(s)}\|^2\le L_i^{(s)}\left(\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2+\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2\right) \tag{C.34}$$

$$B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\ge\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)})-\nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}} \tag{C.35}$$

Therefore,

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\left(B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})-B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\right) \\ &\le\frac{1}{n}\sum_{i=1}^{n}\left(L_i^{(s)}\left(\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2+\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2\right)-\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)})-\nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}}\right)\end{aligned} \tag{C.36}$$

$$=L^{(s)}\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2+\frac{1}{n}\sum_{i=1}^{n}L_i^{(s)}\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2-\frac{1}{n}\sum_{i=1}^{n}\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i)-\nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}} \tag{C.37}$$

where recall $L^{(s)}=\frac{1}{n}\sum_{i=1}^{n}L_i^{(s)}$. Now bound $\frac{1}{n}\sum_{i=1}^{n}L_i^{(s)}\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2$ as follows:

$$\frac{1}{n}\sum_{i=1}^{n} L_i^{(s)}\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2=\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\|\mathbf{x}_i^{(s)}-\mathbf{x}_1^{(s)}\|^2=\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\eta^2\Big\|\sum_{j=1}^{i-1}(\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})+\rho_j^{(s)})\Big\|^2 \tag{C.38}$$

$$=\eta^2\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}(\mathbf{z})+\nabla f_j^{(s)}(\mathbf{z})-\nabla f_j(\mathbf{z}) \tag{C.39}$$

$$+\nabla f_j(\mathbf{z})-\nabla f_j(\mathbf{x}^*)+\nabla f_j(\mathbf{x}^*)+\rho_j^{(s)}\Big)\Big\|^2$$

$$\le\eta^2\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\Big(5\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}(\mathbf{z})\Big)\Big\|^2+5\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j^{(s)}(\mathbf{z})-\nabla f_j(\mathbf{z})\Big)\Big\|^2 \tag{C.40}$$

$$+\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j(\mathbf{z})-\nabla f_j(\mathbf{x}^*)\Big)\Big\|^2+5\Big\|\sum_{j=1}^{i-1}\nabla f_j(\mathbf{x}^*)\Big\|^2+5\Big\|\sum_{j=1}^{i-1}\rho_j^{(s)}\Big\|^2\Big)$$

Note that

$$\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}(\mathbf{z})\Big)\Big\|^2 \tag{C.41}$$

$$\le\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}(i-1)\sum_{j=1}^{i-1}\Big\|\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}(\mathbf{z})\Big\|^2=\frac{1}{n}\sum_{j=1}^{n-1}\Big(\sum_{i=j+1}^{n} L_i^{(s)}(i-1)\Big)\Big\|\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}($$

$$\le\frac{1}{n}\sum_{j=1}^{n-1} n^2L^{(s)}\Big\|\nabla f_j^{(s)}(\mathbf{x}_j^{(s)})-\nabla f_j^{(s)}(\mathbf{z})\Big\|^2\le\sum_{i=1}^{n} nL^{(s)}\Big\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)})-\nabla f_i^{(s)}(\mathbf{z})\Big\|^2 \tag{C.42}$$

By Assumption 4.3.7,

$$\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}\Big\|\sum_{j=1}^{i-1}\Big(\nabla f_j^{(s)}(\mathbf{z})-\nabla f_j(\mathbf{z})\Big)\Big\|^2\le\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)}(C_{i-1}^{(s)})^2\le\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 \tag{C.43}$$

and

$$\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)} \Big\| \sum_{j=1}^{i-1} \Big(\nabla f_j(\mathbf{z}) - \nabla f_j(\mathbf{x}^*) \Big) \Big\|^2 \tag{C.44}$$

$$\overset{(a)}{\leq} \frac{1}{n}\sum_{i=2}^{n} L_i^{(s)} \cdot 2 \Big(\sum_{j=1}^{i-1} L_j^{(s)} \Big) \Big(\sum_{l=1}^{i-1} B_{f_l}(\mathbf{z}, \mathbf{x}^*) \Big) = \frac{1}{n}\sum_{i=2}^{n}\sum_{j=1}^{i-1}\sum_{l=1}^{i-1} 2 L_i^{(s)} L_j^{(s)} B_{f_l}(\mathbf{z}, \mathbf{x}^*)$$

$$= \frac{1}{n}\sum_{j=1}^{n-1}\sum_{l=1}^{n-1} \Big(\sum_{i=\max\{j,l\}+1}^{n} 2 L_i^{(s)} \Big) L_j^{(s)} B_{f_l}(\mathbf{z}, \mathbf{x}^*) \leq \frac{1}{n} 2nL^{(s)} \sum_{j=1}^{n-1}\sum_{l=1}^{n-1} L_j^{(s)} B_{f_l}(\mathbf{z}, \mathbf{x}^*) \tag{C.45}$$

$$\leq \frac{1}{n} 2n^2 (L^{(s)})^2 \sum_{l=1}^{n-1} B_{f_l}(\mathbf{z}, \mathbf{x}^*) \overset{(b)}{\leq} \frac{1}{n} 2n^2 (L^{(s)})^2 \sum_{l=1}^{n} B_{f_l}(\mathbf{z}, \mathbf{x}^*) \tag{C.46}$$

$$= 2n^2 (L^{(s)})^2 B_F(\mathbf{z}, \mathbf{x}^*) \tag{C.47}$$

where (a) is by Lemma C.1.1 and (b) is due to $B_{f_n}(\mathbf{z}, \mathbf{x}^*) \geq 0$.

By Lemma C.1.2,

$$\frac{1}{n}\sum_{i=2}^{n} L_i^{(s)} \Big\| \sum_{j=1}^{i-1} \nabla f_j(\mathbf{x}^*) \Big\|^2 \leq n^2 L^{(s)} \sigma_{any}^2 \tag{C.48}$$

Plugging Eq. C.42, Eq. C.43, Eq. C.47 and Eq. C.48 back to Eq. C.40, there is

$$\frac{1}{n}\sum_{i=1}^{n} L_i^{(s)} \|\mathbf{x}_i^{(s)} - \mathbf{x}_1^{(s)}\|^2 \tag{C.49}$$

$$\leq 5\eta^2 nL^{(s)} \sum_{i=1}^{n} \Big\| \nabla f_i^{(s)}(\mathbf{x}_i^{(s)}) - \nabla f_i^{(s)}(\mathbf{z}) \Big\|^2 + 5\eta^2 \frac{1}{n} \sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2$$

$$+ 10\eta^2 n^2 (L^{(s)})^2 B_F(\mathbf{z}, \mathbf{x}^*) + 5\eta^2 n^2 L^{(s)} \sigma_{any}^2 + 5\eta^2 \frac{1}{n} \sum_{i=2}^{n} L_i^{(s)} \Big\| \sum_{j=1}^{i-1} \rho_j^{(s)} \Big\|^2$$

Plugging Eq. C.49 back to Eq. C.37,

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left(B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})-B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\right) \qquad \text{(C.50)}\\
&\le L^{(s)}\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2-\frac{1}{n}\sum_{i=1}^{n}\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i)-\nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}}\\
&\quad+5\eta^2nL^{(s)}\sum_{i=1}^{n}\left\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)})-\nabla f_i^{(s)}(\mathbf{z})\right\|^2+5\eta^2\frac{1}{n}\sum_{i=1}^{n-1}L_{i+1}^{(s)}(C_i^{(s)})^2\\
&\quad+10\eta^2n^2(L^{(s)})^2B_F(\mathbf{z},\mathbf{x}^*)+5\eta^2n^2L^{(s)}\sigma_{any}^2+5\eta^2\frac{1}{n}\sum_{i=2}^{n}L_i^{(s)}\Big\|\sum_{j=1}^{i-1}\rho_j^{(s)}\Big\|^2
\end{aligned}$$

Taking expectation w.r.t. $\{\rho_j^{(s)}\}_{i=1}^n$ of both sides, and note that

$$\frac{1}{n}\sum_{i=2}^{n}L_i^{(s)}\mathbb{E}\Big\|\sum_{j=1}^{i-1}\rho_j^{(s)}\Big\|^2=\frac{1}{n}\sum_{i=2}^{n}L_i^{(s)}(i-1)d(\sigma^{(s)})^2\le\frac{1}{n}\sum_{i=1}^{n}L_i^{(s)}nd(\sigma^{(s)})^2=L^{(s)}nd(\sigma^{(s)})^2 \qquad \text{(C.51)}$$

We have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)},\mathbf{x}_i^{(s)})\right]-\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z},\mathbf{x}_i^{(s)})\right]\right) \qquad \text{(C.52)}\\
&\le L^{(s)}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2\right]-\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i)-\nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}}\right]\\
&\quad+5\eta^2nL^{(s)}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)})-\nabla f_i^{(s)}(\mathbf{z})\right\|^2\right]+5\eta^2\frac{1}{n}\sum_{i=1}^{n-1}L_{i+1}^{(s)}(C_i^{(s)})^2\\
&\quad+10\eta^2n^2(L^{(s)})^2\mathbb{E}\left[B_F(\mathbf{z},\mathbf{x}^*)\right]+5\eta^2n^2L^{(s)}\sigma_{any}^2+5\eta^2L^{(s)}nd(\sigma^{(s)})^2
\end{aligned}$$

If we set the learning rate η such that

$$5\eta^2nL^{(s)}\le\frac{1}{2nL^{(s)*}}\Rightarrow\eta\le\frac{1}{n\sqrt{10L^{(s)}L^{(s)*}}}\le\frac{1}{n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}}}$$

then

$$5\eta^2 nL^{(s)} \sum_{i=1}^{n} \mathbb{E}\left[\left\|\nabla f_i^{(s)}(\mathbf{x}_i^{(s)}) - \nabla f_i^{(s)}(\mathbf{z})\right\|^2\right] - \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\frac{\|\nabla f_i^{(s)}(\mathbf{x}_i) - \nabla f_i^{(s)}(\mathbf{z})\|^2}{2L_i^{(s)}}\right] \leq 0$$

and so

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{x}_1^{(s+1)}, \mathbf{x}_i^{(s)})\right] - \mathbb{E}\left[B_{f_i^{(s)}}(\mathbf{z}, \mathbf{x}_i^{(s)})\right]\right) \\ &\leq L^{(s)}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)} - \mathbf{x}_1^{(s)}\|^2\right] + 5\eta^2\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 \\ &\quad + 10\eta^2 n^2 (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}, \mathbf{x}^*)\right] + 5\eta^2 n^2 L^{(s)}\sigma_{any}^2 + 5\eta^2 L^{(s)} nd\sigma^2 \end{aligned} \tag{C.53}$$

□

Lemma C.1.6 (One Epoch Convergence)**.** *Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $s \in [K]$ and $\forall \mathbf{z} \in \mathbb{R}^d$, if $\eta \leq \frac{1}{n\sqrt{10\max\{L,\widetilde{L}\},\max\{L^*,\widetilde{L}^*\}}}$ Algorithm 2 guarantees*

$$\begin{aligned} &\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \\ &\leq \frac{1}{2\eta n}\left(\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2\right] - \mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right]\right) + \left(\frac{L_H^{(s)} + \beta}{2} - \frac{\mu_\psi}{2}\right)\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right] \\ &\quad + \frac{1}{2\beta}(C^{(s)})^2 + 5\eta^2\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 + 5\eta^2 L^{(s)} nd(\sigma^{(s)})^2 \\ &\quad + 10\eta^2 n^2 (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}, \mathbf{x}^*)\right] + 5\eta^2 n^2 L^{(s)}\sigma_{any}^2 \end{aligned} \tag{C.54}$$

where the expectation is taken w.r.t. the injected noise and $L^{(s)} \in \{L, \widetilde{L}\}$.

Proof of Lemma C.1.6. By Lemma C.1.4 and Lemma C.1.5, for any $s \in [K]$ and $\forall \mathbf{z} \in \mathbb{R}^d$, if $\eta \leq \frac{1}{n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}}}$,

$$\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \tag{C.55}$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] \\
&\quad + \frac{\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s)}\|^2\right]}{2n\eta} - (\frac{1}{2n\eta}+\frac{\mu_\psi}{2})\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s+1)}\|^2\right] - \frac{1}{2n\eta}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2\right] \\
&\quad + L^{(s)}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{x}_1^{(s)}\|^2\right] + 5\eta^2\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 \\
&\quad + 10\eta^2n^2(L^{(s)})^2\mathbb{E}\left[B_F(\mathbf{z},\mathbf{x}^*)\right] + 5\eta^2n^2L^{(s)}\sigma_{any}^2 + 5\eta^2L^{(s)}nd(\sigma^{(s)})^2 \\
&\leq \mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] \qquad \text{(C.56)} \\
&\quad + \frac{\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s)}\|^2\right]}{2n\eta} - (\frac{1}{2n\eta}+\frac{\mu_\psi}{2})\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s+1)}\|^2\right] \\
&\quad + 5\eta^2\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 + 10\eta^2n^2(L^{(s)})^2\mathbb{E}\left[B_F(\mathbf{z},\mathbf{x}^*)\right] + 5\eta^2n^2L^{(s)}\sigma_{any}^2 + 5\eta^2L^{(s)}nd(\sigma^{(s)})^2
\end{aligned}
$$

For any $\beta > 0$,

$$
\begin{aligned}
\mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] &= \mathbb{E}\left[H(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[H(\mathbf{z})\right] - \mathbb{E}\left[\langle\nabla H(\mathbf{z}), \mathbf{x}_1^{(s+1)}-\mathbf{z}\rangle\right] + \mathbb{E}\left[\langle\nabla H(\mathbf{z}), \mathbf{x}_1^{(s+} \qquad \text{(C.57)} \\
&\overset{\text{(a)}}{\leq} \frac{L_H^{(s)}}{2}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{z}\|^2\right] + \frac{1}{2\beta}(C_n^{(s)})^2 + \frac{\beta}{2}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{z}\|^2\right] \qquad \text{(C.58)} \\
&= \frac{L_H^{(s)}+\beta}{2}\mathbb{E}\left[\|\mathbf{x}_1^{(s+1)}-\mathbf{z}\|^2\right] + \frac{1}{2\beta}(C_n^{(s)})^2 \qquad \text{(C.59)}
\end{aligned}
$$

where (a) is by Assumption 4.3.6, 4.3.7 and Young's inequality.

Hence, plugging Eq. C.59 back to Eq. C.56, there is

$$
\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \qquad \text{(C.60)} \\
&\leq \frac{1}{2\eta n}\Big(\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s)}\|^2\right] - \mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s+1)}\|^2\right]\Big) + \Big(\frac{L_H^{(s)}+\beta}{2} - \frac{\mu_\psi}{2}\Big)\mathbb{E}\left[\|\mathbf{z}-\mathbf{x}_1^{(s+1)}\|^2\right] \\
&\quad + \frac{1}{2\beta}(C^{(s)})^2 + 5\eta^2\frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)}(C_i^{(s)})^2 + 5\eta^2L^{(s)}nd(\sigma^{(s)})^2 \\
&\quad + 10\eta^2n^2(L^{(s)})^2\mathbb{E}\left[B_F(\mathbf{z},\mathbf{x}^*)\right] + 5\eta^2n^2L^{(s)}\sigma_{any}^2
\end{aligned}
$$

□

C.1.3 Convergence Across Epochs

Lemma C.1.7. *Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $k \in [K]$ and $\beta > 0$, if $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$ and $\eta \leq \frac{1}{n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}}}$, Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \qquad \text{(C.61)}\\
&\leq \frac{1}{2\eta nk}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 5\eta^2 n^2 \sigma_{any}^2 \sum_{s=1}^{k} v_s L^{(s)} + 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{l=2}^{k} v_{l-1}\mathbb{E}\left[B_F(\mathbf{x}_1^{(s)}, \mathbf{x}^*)\right]\\
&\quad + \frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2
\end{aligned}$$

where the expectation is take w.r.t. the injected noise $\rho_i^{(s)}, \forall s \in [k], i \in [n]$ and $L^{(s)} = \{L, \widetilde{L}\}$.

Proof. If $\mu_\psi \geq L_H^{(s)} + \beta$, $\forall s \in [k]$, then for any $s \in [K]$, the learning rate η satisfies the condition in Lemma C.1.6 and so by this lemma,

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z})\right] \qquad \text{(C.62)}\\
&\leq \frac{1}{2\eta n}\left(\mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s)}\|^2\right] - \mathbb{E}\left[\|\mathbf{z} - \mathbf{x}_1^{(s+1)}\|^2\right]\right) + 10\eta^2 n^2 (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}, \mathbf{x}^*)\right] + 5\eta^2 n^2 L^{(s)} \sigma_{any}^2\\
&\quad + \frac{1}{2\beta}(C^{(s)})^2 + 5\eta^2 \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 L^{(s)} nd (\sigma^{(s)})^2
\end{aligned}$$

Fix $k \in [K]$, and define the non-decreasing sequence

$$v_s = \frac{1}{k+1-s}, \forall s \in [k], \quad v_0 = v_1 = \frac{1}{k}$$

and the auxiliary points

$$\mathbf{z}^{(0)} = \mathbf{x}^*, \quad \mathbf{z}^{(s)} = \left(1 - \frac{v_{s-1}}{v_s}\right)\mathbf{x}_1^{(s)} + \frac{v_{s-1}}{v_s}\mathbf{z}^{(s-1)}, \forall s \in [k] \qquad \text{(C.63)}$$

Equivalently, $\mathbf{z}^{(s)}$ can be re-written as

$$\mathbf{z}^{(s)} = \frac{v_0}{v_s}\mathbf{x}^* + \sum_{l=1}^{s} \frac{v_l - v_{l-1}}{v_s}\mathbf{x}_1^{(l)}, \forall s \in [0] \cup [k] \tag{C.64}$$

For any $s \in [k]$, setting $\mathbf{z} = \mathbf{z}^{(s)}$ in Eq. C.65 leads to

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z}^{(s)})\right] \\
&\leq \frac{1}{2\eta n}\Big(\mathbb{E}\left[\|\mathbf{z}^{(s)} - \mathbf{x}_1^{(s)}\|^2\right] - \mathbb{E}\left[\|\mathbf{z}^{(s)} - \mathbf{x}_1^{(s+1)}\|^2\right]\Big) + 10\eta^2 n^2 (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}^{(s)}, \mathbf{x}^*)\right] + 5\eta^2 n^2 L^{(s)} \sigma_{any}^2 \\
&\quad + \frac{1}{2\beta}(C^{(s)})^2 + 5\eta^2 \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 L^{(s)} n d (\sigma^{(s)})^2
\end{aligned} \tag{C.65}$$

Note that by Eq. C.63,

$$\|\mathbf{z}^{(s)} - \mathbf{x}_1^{(s)}\|^2 = \frac{v_{s-1}^2}{v_s^2}\|\mathbf{z}^{(s-1)} - \mathbf{x}_1^{(s)}\|^2 \leq \frac{v_{s-1}}{v_s}\|\mathbf{z}^{(s-1)} - \mathbf{x}^{(s)}\|^2 \tag{C.66}$$

where the last inequality is due to $v_{s-1} \leq v_s$. Hence,

$$\begin{aligned}
&v_s \cdot \Big(E\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z}^{(s)})\right]\Big) \\
&\leq \frac{1}{2\eta n}\Big(v_{s-1}\mathbb{E}\left[\|\mathbf{z}^{(s-1)} - \mathbf{x}^{(s)}\|^2\right] - \mathbb{E}\left[\|\mathbf{z}^{(s)} - \mathbf{x}_1^{(s+1)}\|^2\right]\Big) + 10 v_s \eta^2 n^2 (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}^{(s)}, \mathbf{x}^*)\right] + 5 v_s \eta^2 n^2 1 \\
&\quad + \frac{1}{2\beta} v_s (C^{(s)})^2 + 5 v_s \eta^2 \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5 v_s \eta^2 L^{(s)} n d (\sigma^{(s)})^2
\end{aligned} \tag{C.67}$$

Summing Eq. C.67 from $s = 1$ to k to obtain

$$\begin{aligned}
&\sum_{s=1}^{k} v_s \cdot \Big(E\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z}^{(s)})\right]\Big) \\
&\leq \frac{1}{2\eta n}\Big(v_0 \mathbb{E}\left[\|\mathbf{z}^{(0)} - \mathbf{x}_1^{(1)}\|^2\right] - v_k \mathbb{E}\left[\|\mathbf{z}^{(k)} - \mathbf{x}_1^{(k+1)}\|^2\right]\Big) \\
&\quad + 10\eta^2 n^2 \sum_{s=1}^{k} v_s (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}^{(s)}, \mathbf{x}^*)\right] + 5\eta^2 n^2 \sigma_{any}^2 \sum_{s=1}^{k} v_s L^{(s)}
\end{aligned} \tag{C.68}$$

$$+\frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2$$

Note since $\|\mathbf{z}^{(k)} - \mathbf{x}_1^{(k+1)}\|^2 \geq 0$ and $\mathbf{z}^{(0)} = \mathbf{x}^*, v_0 = \frac{1}{k}$,

$$\frac{1}{2\eta n}\Big(v_0 \mathbb{E}\left[\|\mathbf{z}^{(0)} - \mathbf{x}_1^{(1)}\|^2\right] - v_k \mathbb{E}\left[\|\mathbf{z}^{(k)} - \mathbf{x}_1^{(k+1)}\|^2\right]\Big) \leq \frac{1}{2\eta n k}\mathbb{E}\left[\|\mathbf{x}^* - \mathbf{x}_1^{(1)}\|^2\right] \tag{C.69}$$

By the convexity of $F(\mathbf{x})$ and Eq. C.64,

$$F(\mathbf{z}^{(s)}) \leq \frac{v_0}{v_s} F(\mathbf{x}^*) + \sum_{l=1}^{s} \frac{v_l - v_{l-1}}{v_s} F(\mathbf{x}_1^{(l)}) = F(\mathbf{x}^*) + \sum_{l=1}^{s} \frac{v_l - v_{l-1}}{v_s}\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big) \tag{C.70}$$

which implies

$$\sum_{s=1}^{k} v_s \Big(F(\mathbf{x}_1^{(s+1)}) - F(\mathbf{z}^{(s)})\Big) \tag{C.71}$$

$$\geq \sum_{s=1}^{k} \Big(v_s\Big(F(\mathbf{x}_1^{(s+1)}) - F(\mathbf{x}^*)\Big) - \sum_{l=1}^{s}(v_l - v_{l-1})\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big)\Big)$$

$$\geq \sum_{s=1}^{k} v_s \Big(F(\mathbf{x}_1^{(s+1)}) - F(\mathbf{x}^*)\Big) - \sum_{s=1}^{k}\sum_{l=1}^{s}(v_l - v_{l-1})\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big) \tag{C.72}$$

$$= \sum_{s=1}^{k} v_s \Big(F(\mathbf{x}_1^{(s+1)}) - F(\mathbf{x}^*)\Big) - \sum_{l=1}^{k}(k+1-l)(v_l - v_{l-1})\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big) \tag{C.73}$$

$$= v_k \Big(F(\mathbf{x}_1^{(k+1)}) - F(\mathbf{x}^*)\Big) - \sum_{s=1}^{k-1} \frac{1}{k+1-s}\Big(F(\mathbf{x}_1^{(s+1)}) - F(\mathbf{x}^*)\Big) \tag{C.74}$$

$$- k(v_1 - v_0)\Big(F(\mathbf{x}_1^{(1)}) - F(\mathbf{x}^*)\Big) - \sum_{l=2}^{k}(k+1-l)\Big(\frac{1}{k+1-l} - \frac{1}{k+2-l}\Big)\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big)$$

$$= v_k \Big(F(\mathbf{x}_1^{(k+1)}) - F(\mathbf{x}^*)\Big) - \sum_{s=2}^{k} \frac{1}{k+2-s}\Big(F(\mathbf{x}_1^{(s)}) - F(\mathbf{x}^*)\Big) \tag{C.75}$$

$$- k(v_1 - v_0)\Big(F(\mathbf{x}_1^{(1)}) - F(\mathbf{x}^*)\Big) - \sum_{l=2}^{k} \frac{1}{k+2-l}\Big(F(\mathbf{x}_1^{(l)}) - F(\mathbf{x}^*)\Big)$$

By definition, $v_1 = v_0$ and $v_k = 1$, hence, taking expectation of both sides,

$$\sum_{s=1}^{k} v_s \Big(\mathbb{E}\left[F(\mathbf{x}_1^{(s+1)})\right] - \mathbb{E}\left[F(\mathbf{z}^{(s)})\right] \Big) \geq \mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \tag{C.76}$$

To bound $10\eta^2 n^2 \sum_{s=1}^{k} v_s (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}^{(s)}, \mathbf{x}^*)\right]$, by the convexity of $B_F(\cdot, \mathbf{x}^*)$ fixing the second argument (due to F being convex), and Eq. C.64,

$$B_F(\mathbf{z}^{(s)}, \mathbf{x}^*) \leq \frac{v_0}{v_s} B_F(\mathbf{x}^*, \mathbf{x}^*) + \sum_{l=1}^{s} \frac{v_l - v_{l-1}}{v_s} B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*) = \sum_{l=1}^{s} \frac{v_l - v_{l-1}}{v_s} B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*) \tag{C.77}$$

which implies

$$10\eta^2 n^2 \sum_{s=1}^{k} v_s (L^{(s)})^2 \mathbb{E}\left[B_F(\mathbf{z}^{(s)}, \mathbf{x}^*)\right] \leq 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{s=1}^{k} \sum_{l=1}^{s} (v_l - v_{l-1}) \mathbb{E}\left[B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*)\right] \tag{C.78}$$

$$= 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{l=1}^{k} (k+1-l)(v_l - v_{l-1}) \mathbb{E}\left[B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*)\right] \tag{C.79}$$

$$= 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{l=2}^{k} (k+1-l)\left(\frac{1}{k+1-l} - \frac{1}{k+2-l}\right) \mathbb{E}\left[B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*)\right] \tag{C.80}$$

$$= 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{l=2}^{k} \frac{1}{k+2-l} \mathbb{E}\left[B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*)\right] \tag{C.81}$$

$$= 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{l=2}^{k} v_{l-1} \mathbb{E}\left[B_F(\mathbf{x}_1^{(l)}, \mathbf{x}^*)\right] \tag{C.82}$$

Plugging Eq. C.69, Eq. C.76 and Eq. C.82 back to Eq. C.68, we have

$$\mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \tag{C.83}$$

$$\leq \frac{1}{2\eta n k} \mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 5\eta^2 n^2 \sigma_{any}^2 \sum_{s=1}^{k} v_s L^{(s)} + 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\} \sum_{s=2}^{k} v_{s-1} \mathbb{E}\left[B_F(\mathbf{x}_1^{(s)}, \mathbf{x}^*)\right]$$

$$+\frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2$$

□

Theorem C.1.8 (Convergence for K Epochs (Re-statement of Theorem 4.3.11))**.** *Under Assumption 4.3.1, 4.3.2, 4.3.6 and 4.3.7, for any $k \in [K]$ and $\beta > 0$, if $\mu_\psi \geq L_H^{(s)} + \beta, \forall s \in [k]$ and $\eta \leq \frac{1}{2n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}(1+\log K)}}$, Algorithm 2 guarantees*

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \qquad \text{(C.84)}\\
&\leq \frac{1}{\eta n K}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 10\eta^2 n^2 \sigma_{any}^2 (1+\log K)\max\{L, \widetilde{L}\}\\
&\quad + \frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2
\end{aligned}$$

where the expectation is taken w.r.t. the injected noise.

Proof of Theorem 4.3.11. Taking $\eta \leq \frac{1}{2n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}(1+\log K)}}$ satisfies the condition in Lemma C.1.7, and by this lemma, for any $k \in [K]$,

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \qquad \text{(C.85)}\\
&\leq \frac{1}{2\eta n k}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 5\eta^2 n^2 \sigma_{any}^2 \sum_{s=1}^{k} v_s L^{(s)} + 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\}\sum_{s=2}^{k} v_{s-1}\mathbb{E}\left[B_F(\mathbf{x}_1^{(s)}, \mathbf{x}^*)\right]\\
&\quad + \frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2
\end{aligned}$$

where $v_s = \frac{1}{k+1-s}$.

Note that $\sum_{s=1}^{k} v_s = \sum_{s=1}^{k} \frac{1}{k+1-s} = \sum_{s=1}^{k} \frac{1}{s} \leq 1 + \log k$, and so

$$\eta^2 n^2 \sigma_{any}^2 \sum_{s=1}^{k} L^{(s)} \leq \eta^2 n^2 \sigma_{any}^2 (1+\log k)\max\{L, \widetilde{L}\}$$

Hence,

$$\mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \tag{C.86}$$
$$\leq \frac{1}{2\eta n k}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 5\eta^2 n^2 \sigma_{any}^2 (1+\log k)\max\{L, \widetilde{L}\} + 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\}\sum_{s=2}^{k} v_{s-1}\mathbb{E}\left[B_F(\mathbf{x}\right.$$
$$+ \frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2$$

The definition of $\mathbf{x} \in \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$ implies $\exists \nabla\psi(\mathbf{x}^*) \in \partial\psi(\mathbf{x}^*)$ such that $\nabla F(\mathbf{x}^*) + \nabla\psi(\mathbf{x}^*) = \mathbf{0}$. Thus, for any $k \in [K]$,

$$\mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \geq \mathbb{E}\left[F(\mathbf{x}_1^{(k+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] - \mathbb{E}\left[\langle \nabla F(\mathbf{x}^*) + \nabla\psi(\mathbf{x}^*), \mathbf{x}_1^{(k+1)} - \mathbf{x}^*\rangle\right] \tag{C.87}$$
$$= \mathbb{E}\left[B_F(\mathbf{x}_1^{(k+1)}, \mathbf{x}^*)\right] + \mathbb{E}\left[B_\psi(\mathbf{x}_1^{(k+1)}, \mathbf{x}^*)\right] \geq \mathbb{E}\left[B_F(\mathbf{x}_1^{(k+1)}, \mathbf{x}^*)\right] \tag{C.88}$$

which implies

$$\mathbb{E}\left[B_F(\mathbf{x}_1^{(k+1)}, \mathbf{x}^*)\right] \tag{C.89}$$
$$\leq \frac{1}{2\eta n k}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right] + 5\eta^2 n^2 \sigma_{any}^2 (1+\log k)\max\{L, \widetilde{L}\} + 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\}\sum_{s=2}^{k} v_{s-1}\mathbb{E}\left[B_F(\mathbf{x}\right.$$
$$+ \frac{1}{2\beta}\sum_{s=1}^{k} v_s (C^{(s)})^2 + 5\eta^2 \cdot \sum_{s=1}^{k} v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd \sum_{s=1}^{k} v_s L^{(s)} (\sigma^{(s)})^2$$

Now we apply Lemma C.1.3 with $d_{k+1} = \begin{cases} \mathbb{E}\left[B_F(\mathbf{x}_1^{(k+1)}, \mathbf{x}^*)\right] & k \in [K-1] \\ \mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] & k = K \end{cases}$
$a = \frac{1}{2\eta n}\mathbb{E}\left[\|\mathbf{x}_1^{(1)} - \mathbf{x}^*\|^2\right]$, $b = 5\eta^2 n^2 \sigma_{any}^2 (1+\log k)\max\{L, \widetilde{L}\}$, $c = 10\eta^2 n^2 \max\{L^2, \widetilde{L}^2\}$ and $r^{(s)} = \frac{1}{2\beta} v_s (C^{(s)})^2 + 5\eta^2 v_s \frac{1}{n}\sum_{i=1}^{n-1} L_{i+1}^{(s)} (C_i^{(s)})^2 + 5\eta^2 nd v_s L^{(s)} (\sigma^{(s)})^2$, to obtain

$$\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right] - \mathbb{E}\left[F(\mathbf{x}^*)\right] \tag{C.90}$$

$$\leq \Big(\frac{1}{2\eta nK}\mathbb{E}\left[\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^2\right]+5\eta^2n^2\sigma_{any}^2(1+\log K)\max\{L,\widetilde{L}\}\Big)\sum_{i=0}^{K-1}\Big(20\eta^2n^2\max\{L^2,\widetilde{L}^2\}(1+\log K)\Big)^i$$
$$+\frac{1}{2\beta}\sum_{s=1}^{k}v_s(C^{(s)})^2+5\eta^2\cdot\sum_{s=1}^{k}v_s\frac{1}{n}\sum_{i=1}^{n-1}L_{i+1}^{(s)}(C_i^{(s)})^2+5\eta^2nd\sum_{s=1}^{k}v_sL^{(s)}(\sigma^{(s)})^2$$

By setting $\eta\leq\frac{1}{2n\sqrt{10\max\{L,\widetilde{L}\}\cdot\max\{L^*,\widetilde{L}^*\}(1+\log K)}}$, there is

$$\sum_{i=0}^{K-1}\Big(20\eta^2n^2\max\{L^2,\widetilde{L}^2\}(1+\log K)\Big)^i\leq\sum_{i=0}^{K-1}\Big(\frac{\max\{L^2,\widetilde{L}^2\}x(1+\log K)}{2\max\{L,\widetilde{L}\}\max\{L^*,\widetilde{L}^*\}(1+\log K)}\Big)^i\leq\sum_{i=0}^{\infty}\frac{1}{2^i}=2 \tag{C.91}$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left[F(\mathbf{x}_1^{(K+1)})\right]-\mathbb{E}\left[F(\mathbf{x}^*)\right] \qquad\qquad (C.92)\\ &\leq\frac{1}{\eta nK}\mathbb{E}\left[\|\mathbf{x}_1^{(1)}-\mathbf{x}^*\|^2\right]+10\eta^2n^2\sigma_{any}^2(1+\log K)\max\{L,\widetilde{L}\}\\ &\quad+\frac{1}{2\beta}\sum_{s=1}^{k}v_s(C^{(s)})^2+5\eta^2\cdot\sum_{s=1}^{k}v_s\frac{1}{n}\sum_{i=1}^{n-1}L_{i+1}^{(s)}(C_i^{(s)})^2+5\eta^2nd\sum_{s=1}^{k}v_sL^{(s)}(\sigma^{(s)})^2\end{aligned}$$

□

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[3] Jonathan Ullman Adam Smith. Privacy in statistics and machine learning, lecture 12: Private gradient descent, 2021. URL https://dpcourse.github.io/2021-spring/lecnotes-web/lec-12-GD.pdf.

[4] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 557–563, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132597. URL https://doi.org/10.1145/1132516.1132597.

[5] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[6] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

[7] Jason Altschuler and Kunal Talwar. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=pDUYkwrx__w.

[8] Peter Arbenz, Walter Gander, and Gene H. Golub. Restricted rank modification of the symmetric eigenvalue problem: Theoretical considerations. *Linear Algebra and its Applications*, 104:75–95, 1988. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(88)90309-6. URL https://www.sciencedirect.com/science/article/pii/0024379588903096.

[9] Oleg Balabanov, Matthias Beaupère, Laura Grigori, and Victor Lederer. Block subsampled randomized Hadamard transform for low-rank approximation on distributed architectures. working paper or preprint, October 2022. URL https://inria.hal.science/hal-03828607.

[10] Maria-Florina F Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general topologies. *Advances in neural information processing systems*, 26, 2013.

[11] Leighton Pate Barnes, Huseyin A Inan, Berivan Isik, and Ayfer Özgür. rtop-k: A statistical estimation approach to distributed sgd. *IEEE Journal on Selected Areas in Information Theory*, 1(3):897–907, 2020.

[12] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, page 464–473, USA, 2014. IEEE Computer Society. ISBN 9781479965175. doi: 10.1109/FOCS.2014.56. URL https://doi.org/10.1109/FOCS.2014.56.

[13] Raef Bassily, Om Thakkar, and Abhradeep Thakurta. Model-agnostic private learning via stability. *arXiv preprint arXiv:1803.05101*, 2018.

[14] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[16] Adam Block, Mark Bun, Rathin Desai, Abhishek Shetty, and Steven Wu. Oracle-efficient differentially private learning with public data, 2024. URL https://arxiv.org/abs/2402.09483.

[17] 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.

[18] Christos Boutsidis and Alex Gittens. Improved matrix algorithms via the subsampled randomized hadamard transform, 2013.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[20] Wei-Ning Chen, Peter Kairouz, and Ayfer Ozgur. Breaking the communication-privacy-accuracy trilemma. *Advances in Neural Information Processing Systems*, 33:3312–3324, 2020.

[21] Christopher A. Choquette-Choo, H. Brendan McMahan, Keith Rush, and Abhradeep Thakurta. Multi-epoch matrix factorization mechanisms for private machine learning, 2023. URL https://arxiv.org/abs/2211.06530.

[22] Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How private are DP-SGD implementations? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8904–8918. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chua24a.html.

[23] Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Scalable DP-SGD: Shuffling vs. poisson subsampling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6gMnj9oc6d.

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

[25] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging, 2024. URL https://arxiv.org/abs/2206.05091.

[26] Peter Davies, Vijaykrishna Gurunathan, Niusha Moshrefi, Saleh Ashkboos, and Dan Alistarh. New bounds for distributed mean estimation and variance reduction, 2021.

[27] Jinshuo Dong, David Durfee, and Ryan Rogers. Optimal differential privacy composition for exponential mechanisms. In *International Conference on Machine Learning*, pages 2597–2606. PMLR, 2020.

[28] David Durfee and Ryan M Rogers. Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *Conference On Learning Theory*, pages 1693–1702. PMLR, 2018.

[30] 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, 2009.

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

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

[33] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 1054–1067, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329576. doi: 10.1145/2660267.2660348. URL https://doi.org/10.1145/2660267.2660348.

[34] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[35] Huang Fang, Xiaoyun Li, Chenglin Fan, and Ping Li. Improved convergence of differential private SGD with gradient clipping. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FRLswckPXQ5.

[36] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, October 2018. doi: 10.1109/focs.2018.00056. URL http://dx.doi.org/10.1109/FOCS.2018.00056.

[37] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532. IEEE, 2018.

[38] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962. doi: 10.1109/TIT.1962.1057683.

[39] Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar. vqsgd: Vector quantized stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics*, pages 2197–2205. PMLR, 2021.

[40] Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of dis-

tributed statistical estimation and dimensionality. *Advances in Neural Information Processing Systems*, 27, 2014.

[41] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning?, 2020. URL https://arxiv.org/abs/2003.14053.

[42] Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, 62(2):925–951, 2015.

[43] Gene H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973. ISSN 00361445. URL http://www.jstor.org/stable/2028604.

[44] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[45] Ming Gu and Stanley C. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1266–1276, 1994. doi: 10.1137/S089547989223924X. URL https://doi.org/10.1137/S089547989223924X.

[46] John A Gubner. Distributed estimation and quantization. *IEEE Transactions on Information Theory*, 39(4):1456–1459, 1993.

[47] Farzin Haddadpour, Belhal Karimi, Ping Li, and Xiaoyun Li. Fedsketch: Communication-efficient and private federated learning via sketching. *arXiv preprint arXiv:2008.04975*, 2020.

[48] Mostafa Haghir Chehreghani. Subsampled randomized hadamard transform for regression of dynamic graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2045–2048, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412158. URL https://doi.org/10.1145/3340531.3412158.

[49] Naoise Holohan, Douglas J. Leith, and Oliver Mason. Optimal differentially private mechanisms for randomised response. *IEEE Transactions on Information Forensics and Security*, 12(11):2726–2735, November 2017. ISSN 1556-6021. doi: 10.1109/tifs.2017.2718487. URL http://dx.doi.org/10.1109/TIFS.2017.2718487.

[50] Samuel Horváth and Peter Richtarik. A better alternative to error feedback for communication-efficient distributed learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vYVI1CHPaQg.

[51] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Ion Stoica, Raman Arora, et al. Communication-efficient distributed sgd with sketching. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Divyansh Jhunjhunwala, Ankur Mallick, Advait Harshal Gadhikar, Swanand Kadhe, and Gauri Joshi. Leveraging spatial and temporal correlations in sparsified mean estimation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=BKeJmkspvc.

[53] Junjie Jia and Wanyong Qiu. Research on an ensemble classification algorithm based on differential privacy. *IEEE Access*, 8:93499–93513, 2020. doi: 10.1109/ACCESS.2020.2995058.

[54] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[55] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling, 2021. URL https://arxiv.org/abs/2103.00039.

[56] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[57] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2021. URL https://arxiv.org/abs/1912.04977.

[58] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019.

[59] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized sgd with changing topology and local updates, 2021. URL https://arxiv.org/abs/2003.10422.

[60] Jakub Konečnỳ and Peter Richtárik. Randomized distributed mean estimation: Accuracy vs. communication. *Frontiers in Applied Mathematics and Statistics*, 4:62, 2018.

[61] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[62] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[63] Jonathan Lacotte and Mert Pilanci. Optimal randomized first-order methods for least-squares problems, 2020.

[64] Jonathan Lacotte, Sifan Liu, Edgar Dobriban, and Mert Pilanci. Optimal iterative sketching with the subsampled randomized hadamard transform, 2020.

[65] Zijian Lei and Liang Lan. Improved subsampled randomized hadamard transform for linear svm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4519–4526, 2020.

[66] Kai Liang and Youlong Wu. Improved communication efficiency for distributed mean estimation with side information. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 3185–3190. IEEE, 2021.

[67] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 298–309, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi: 10.1145/3313276.3316377. URL https://doi.org/10.1145/3313276.3316377.

[68] Zhongfeng Liu, Yun Li, and Wei Ji. Differential private ensemble feature selection. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2018. doi: 10.1109/IJCNN.2018.8489308.

[69] Zijian Liu and Zhengyuan Zhou. On the last-iterate convergence of shuffling gradient methods, 2024. URL https://arxiv.org/abs/2403.07723.

[70] Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/

file/621bf66ddb7c962aa0d22ac97d69b793-Paper.pdf.

[71] Prathamesh Mayekar, Ananda Theertha Suresh, and Himanshu Tyagi. Wyner-ziv estimators: Efficient distributed mean estimation with side-information. In *International Conference on Artificial Intelligence and Statistics*, pages 3502–3510. PMLR, 2021.

[72] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[73] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

[74] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023. URL https://arxiv.org/abs/1602.05629.

[75] Gregory T. Minton and Eric Price. Improved concentration bounds for count-sketch, 2013.

[76] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Ali Jalali, Dean Tullsen, and Hadi Esmaeilzadeh. Shredder: Learning noise distributions to protect inference privacy. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 3–18, 2020.

[77] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, August 2017. doi: 10.1109/csf.2017.11. URL http://dx.doi.org/10.1109/CSF.2017.11.

[78] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Proximal and federated random reshuffling, 2021. URL https://arxiv.org/abs/2102.06704.

[79] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements, 2021. URL https://arxiv.org/abs/2006.05988.

[80] Moni Naor, Kobbi Nissim, Uri Stemmer, and Chao Yan. Private everlasting prediction. *arXiv preprint arXiv:2305.09579*, 2023.

[81] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*, 2020.

[82] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual*

ACM symposium on Theory of computing, pages 75–84, 2007.

[83] Matthew Nokleby and Waheed U Bajwa. Stochastic optimization from distributed streaming data in rate-limited networks. *IEEE transactions on signal and information processing over networks*, 5(1):152–167, 2018.

[84] Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Time-correlated sparsification for communication-efficient federated learning. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 461–466. IEEE, 2021.

[85] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-70L8lpp9DF.

[86] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1610.05755.

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

[88] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[89] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265. PMLR, 2020.

[90] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[91] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[92] Shaohuai Shi, Xiaowen Chu, Ka Chun Cheung, and Simon See. Understanding top-k sparsification in distributed deep learning. *arXiv preprint arXiv:1911.08772*, 2019.

[93] Nir Shlezinger, Mingzhe Chen, Yonina C Eldar, H Vincent Poor, and Shuguang Cui. Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 69:500–514, 2020.

[94] Amin Shokrollahi. Fountain codes. *Iee Proceedings-communications - IEE*

PROC-COMMUN, 152, 01 2005.

[95] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] Uri Stemmer. Private truly-everlasting robust-prediction. *arXiv preprint arXiv:2401.04311*, 2024.

[97] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in Neural Information Processing Systems*, 31, 2018.

[98] Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International conference on machine learning*, pages 3329–3337. PMLR, 2017.

[99] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed mean estimation with limited communication, 2017. URL https://arxiv.org/abs/1611.00429.

[100] Ananda Theertha Suresh, Ziteng Sun, Jae Ro, and Felix Yu. Correlated quantization for distributed mean estimation and optimization. In *International Conference on Machine Learning*, pages 20856–20876. PMLR, 2022.

[101] Ananda Theertha Suresh, Ziteng Sun, Jae Hun Ro, and Felix Yu. Correlated quantization for distributed mean estimation and optimization, 2022. URL https://arxiv.org/abs/2203.04925.

[102] Dan Teng, Xiaowei Zhang, Li Cheng, and Delin Chu. Least squares approximation via sparse subsampled randomized hadamard transform. *IEEE Transactions on Big Data*, 8(2):446–457, 2022. doi: 10.1109/TBDATA.2020.2972887.

[103] 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.

[104] Rohit Tripathy, Ilias Bilionis, and Marcial Gonzalez. Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191–223, 2016.

[105] Joel A. Tropp. Improved analysis of the subsampled randomized hadamard transform, 2011.

[106] Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-57048-8_7. URL https://doi.org/10.1007/978-3-319-57048-8_7.

[107] Laurens van der Maaten and Awni Hannun. The trade-offs of private prediction, 2020.

[108] Shay Vargaftik, Ran Ben-Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben-Itzhak, and Michael Mitzenmacher. Drive: One-bit distributed mean estimation.

Advances in Neural Information Processing Systems, 34:362–377, 2021.

[109] Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben-Itzhak, and Michael Mitzenmacher. Eden: Communication-efficient and robust distributed mean estimation for federated learning, 2022.

[110] Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben Itzhak, and Michael Mitzenmacher. Eden: Communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning*, pages 21984–22014. PMLR, 2022.

[111] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[112] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. A novel framework for the analysis and design of heterogeneous federated learning. *IEEE Transactions on Signal Processing*, 69:5234–5249, 2021. doi: 10.1109/TSP.2021.3106104.

[113] Jun Wang and Zhi-Hua Zhou. Differentially private learning with small public data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 6219–6226, Apr. 2020. doi: 10.1609/aaai.v34i04.6088. URL https://ojs.aaai.org/index.php/AAAI/article/view/6088.

[114] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[115] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farokhi Farhad, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis, 2019. URL https://arxiv.org/abs/1911.00222.

[116] Kilian Q Weinberger, Fei Sha, and Lawrence K Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 106, 2004.

[117] Blake Woodworth, Konstantin Mishchenko, and Francis Bach. Two losses are better than one: faster optimization using a cheaper proxy. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[118] Ming Xiang and Lili Su. β-stochastic sign SGD: A byzantine resilient and differentially private gradient compressor for federated learning, 2023. URL https://openreview.net/forum?id=oVPqFCI1g7q.

[119] Tao Xiang, Yang Li, Xiaoguo Li, Shigang Zhong, and Shui Yu. Collaborative ensemble learning under differential privacy. *Web Intelligence*, 16:73–87, 03 2018. doi: 10.3233/WEB-180374.

[120] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[121] Jiayuan Ye and Reza Shokri. Differentially private learning needs hidden state (or much faster convergence). In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ipAz7H8pPnI.

[122] Ying-Ying Zhang, Teng-Zhong Rong, and Man-Man Li. Expectation identity for the binomial distribution and its application in the calculations of high-order binomial moments. *Communications in Statistics - Theory and Methods*, 48(22):5467–5476, 2019. doi: 10.1080/03610926.2018.1435818. URL https://doi.org/10.1080/03610926.2018.1435818.

[123] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. *Advances in Neural Information Processing Systems*, 26, 2013.

[124] Mingxun Zhou, Tianhao Wang, T-H. Hubert Chan, Giulia Fanti, and Elaine Shi. Locally differentially private sparse vector aggregation. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 422–439, 2022. doi: 10.1109/SP46214.2022.9833635.

[125] 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.