Privacy Amplification by Iteration for ADMM with (Strongly) Convex Objective Functions

T-H. Hubert Chan[*], **Hao Xie**[*], **Mengshi Zhao**[*]

Department of Computer Science, The University of Hong Kong
hubert@cs.hku.hk, hxie@connect.hku.hk, zmsxsl@connect.hku.hk

Abstract

We examine a private ADMM variant for (strongly) convex objectives which is a primal-dual iterative method. Each iteration has a user with a private function used to update the primal variable, masked by Gaussian noise for local privacy, without directly adding noise to the dual variable. Privacy amplification by iteration explores if noises from later iterations can enhance the privacy guarantee when releasing final variables after the last iteration.

Cyffers et al. explored privacy amplification by iteration for the proximal ADMM variant, where a user's entire private function is accessed and noise is added to the primal variable. In contrast, we examine a private ADMM variant requiring just one gradient access to a user's function, but both primal and dual variables must be passed between successive iterations.

To apply Balle et al.'s coupling framework to the gradient ADMM variant, we tackle technical challenges with novel ideas. First, we address the non-expansive mapping issue in ADMM iterations by using a customized norm. Second, because the dual variables are not masked with any noise directly, their privacy guarantees are achieved by treating two consecutive noisy ADMM iterations as a Markov operator.

Our main result is that the privacy guarantee for the gradient ADMM variant can be amplified proportionally to the number of iterations. For strongly convex objective functions, this amplification exponentially increases with the number of iterations. These amplification results align with the previously studied special case of stochastic gradient descent.

1 Introduction

Alternating direction method of multipliers (Gabay and Mercier 1976) (ADMM) has been designed for convex programs whose objective functions can be decomposed as the sum $\mathfrak{f}(x) + g(y)$ of two convex functions[1], where the *primal* variables x and y are restricted by some linear constraint $Ax + By = c$.

Decomposing the objective function into the sum of two convex functions offers several advantages. Firstly, different optimization algorithms can be applied to each part of the function. Secondly, in a distributed learning setting, the function $\mathfrak{f}$ refers to the *loss functions* from various users which can be optimized in parallel, while the function g refers to a *regularizer* term that can typically be optimized by a central server.

In ADMM, a *dual* variable λ keeps track of how much the linear constraint is violated. The method is an iterative procedure that minimizes some *Lagrangian* function $\mathcal{L}(x, y, \lambda)$, which is defined in terms of $\mathfrak{f}$ and g. In each iteration, the three variables x, y and λ are updated in sequential order. While the primal variables x and y are each updated (while keeping other variables constant) to minimize $\mathcal{L}$, the dual variable λ is updated to encourage the feasibility of the linear constraint. The alternating nature of variable updates makes the method widely adaptable in large-scale distributed contexts (Boyd et al. 2011).

In this paper, we focus on a *stochastic* version of ADMM proposed by (Ouyang et al. 2013), in which the function $\mathfrak{f}$ can be viewed as an expectation of functions sampled from some distribution $\mathcal{D}$. Each iteration t is associated with some user, whose (private) data is some function f_t sampled from $\mathcal{D}$. Instead of directly accessing $\mathfrak{f}$, each iteration t only has access to the corresponding user's function f_t. While the sequence $f_1, f_2, \ldots, f_T$ of sampled functions arises from user data, the function g is publicly known. In one ADMM iteration, f_t is only needed for updating the x variable. In the *proximal* variant, the whole function f_t is used in some optimization step to update x. Instead, we will focus on the more computationally efficient *gradient* variant that uses the first order approximation of f_t (Ouyang et al. 2015; Li and Lin 2019), where only one access to the gradient oracle $\nabla f_t(\cdot)$ is sufficient.

All variants of *differential privacy* (Dwork 2006; Bun and Steinke 2016; Mironov 2017) are based on the principle that a mechanism or procedure achieves its privacy guarantee through the incorporation of randomness. Private variants of ADMM have been considered by adding noises to the variables (Zhang and Zhu 2016). To apply this privacy framework to ADMM, the function f_t is considered as the private input of the user in iteration t. Hence, one possible way (Shang et al. 2021) to achieve *local privacy* (against an adversary that can observe the variables after each iteration) for the user t is to sample some noise N_t, which is added to the result of the gradient oracle oracle $\nabla f_t(\cdot)$. A popu-

[*] These authors contributed equally.

[1] It will be clear soon why we use a different font for $\mathfrak{f}$.

lar choice for sampling N_t is Gaussian noise, for which the Rényi D_α and zero-concentrated D^z divergences (formally explained in Section 3) are suitable to measure the closeness of the resulting output distributions.

In the literature, *privacy amplification* loosely refers to the improvement of privacy analysis for a user using extra sources of randomness other than the noise used for achieving its local privacy. An example is the randomness used in sampling data (Chaudhuri and Mishra 2006; Balle, Barthe, and Gaboardi 2018); for ADMM, this can refer to the randomness in sampling each f_t from $\mathcal{D}$. In applications where data from different users can be processed in any arbitrary order, extra randomness from *shuffling* users' data have been considered (Erlingsson et al. 2019; Cheu et al. 2019; Balle et al. 2019b); for ADMM, this can mean that the order of the users in the iterative process is randomly permuted. Privacy amplification *by iteration* (Feldman et al. 2018) has been proposed to analyze an iterative procedure in which some noise is sampled in each iteration to achieve local privacy for the user in that iteration. The improved privacy analysis is from the perspective of the user from the **first** iteration. The intuition is that by exploiting the extra randomness generated in subsequent iterations, the privacy guarantee against an adversary that observes only the result at the end of the final iteration can be improved. In this paper, we consider privacy amplification by iteration for ADMM; in other words, we consider a deterministic sequence $f_1, f_2, \ldots, f_T$ of functions, where the function f_t is used in iteration t of ADMM, and the only source of randomness is the noise N_t sampled in each iteration t, which is used to mask only the x variable (during access to the gradient oracle).

Loosely speaking, each iteration in the iterative process considered in (Feldman et al. 2018; Balle et al. 2019a) corresponds to a *non-expansive* mapping, and an independent copy of Gaussian noise is added to the result of each iteration before passing to the next iteration. From the perspective of the user from the first iteration, the privacy guarantee of the final output after T iterations, when measured with the D^z-divergence[2], is improved by a multiplicative factor of T.

A recent work (Cyffers, Bellet, and Basu 2023) employed this framework to examine privacy amplification by iteration in the **proximal** variant of ADMM, for the purpose of analyzing privacy leakage to both the adversary and among different users.

Our Contribution. The main purpose of this paper is to apply the approaches in (Feldman et al. 2018; Balle et al. 2019a) to achieve privacy amplification by iteration for the **gradient** variant (that uses only the gradient oracle to update the variable x). When one iteration of ADMM is considered, the proximal variant as considered in (Cyffers, Bellet, and Basu 2023) needs to pass only one variable between successive iterations, while the gradient variant needs to pass both the x and λ variables. However, when one analyzes the transition of variables in the (x, λ)-space, there turns out to be two major technical hurdles, which we give high levels ideas for how we resolve them (where more details are described in Section 4.1).

Non-expansive iteration. In (Feldman et al. 2018; Balle et al. 2019a), it is essential that each iteration involves a non-expansive mapping in the variable space before adding noise. This also applies to the proximal variant (Cyffers, Bellet, and Basu 2023). However, for the ADMM gradient oracle variant, one iteration signifies a transition in the (x, λ)-space, which can be a strictly expanding under the usual norm. We resolve this by creating a custom norm in the (x, λ)-space that allows for privacy analysis in ADMM and ensures non-expansiveness (or contractiveness for strongly convex objectives).

One-step privacy. In (Feldman et al. 2018; Balle et al. 2019a), each iteration masks variables with Gaussian noise, allowing for straightforward privacy guarantees in terms of D^z-divergence. This applies to the proximal ADMM variant (Cyffers, Bellet, and Basu 2023), where noise is added to the x variable and linearly transformed to mask the λ variable. However, passing both x and λ variables is more complex. The sampled noise masks only the $x$variable, leaving the λ variable exposed, resulting in infinite D^z-divergence. Our innovative idea involves considering one step as two noisy ADMM iterations, using independent noises to mask each component of (x, λ).

Our Informal Statements. We show that from the perspective of the user from the first iteration, the final variables after T noisy ADMM iterations achieve privacy amplification in the sense that the D^z-divergence is proportional to $\frac{1}{T}$; for strongly convex objective functions, the privacy amplification is improved to $\frac{L^T}{T}$, for some $0 < L < 1$. The formal results for the general convex case are in Theorem 5.1. The formal statements for the strongly convex case are given in the full version[3].

Privacy for Other Users. We analyze the privacy guarantee from the perspective of the first user to make the presentation clearer. As pointed out in (Feldman et al. 2018), very simple techniques can extend the privacy guarantees to all users: (1) random permutation of all users; or (2) *random stopping*: if there are n users, stop after a random number $R \in [1..n]$ of iterations. (Hence, with constant probability, a user is either not included in the sample, or the number of iterations after it is $\Omega(n)$.) We give the details in the full version.

Convergence Rates. We emphasize that our contribution is to analyze privacy amplification for private variants of ADMM that have already appeared in the literature (Zhang and Zhu 2016; Shang et al. 2021), whose applications and convergence rates have already been analyzed. However, for completeness, we present the tradeoff between utility (measured by the convergence rate) and privacy (measured by the variance of privacy noise) in the full version.

Experimental Results. Despite being primarily theoretical, we conduct experiments on a general Lasso problem. Specifically, we empirically examine the effects of strong convexity and privacy noise magnitude on convergence rates. The details are given in the full version.

Paper Organization. While the most relevant works are

[2] The results in (Feldman et al. 2018; Balle et al. 2019a) are stated equivalently in terms of Rényi divergence.

[3] The full version of the paper is available on arXiv (Chan, Xie, and Zhao 2023).

mentioned in this section, further details on related work are given in the full version. Background on ADMM is given in Section 2 and formal privacy notions are given in Section 3. In Section 4, we give a short review of the previous coupling approach (Balle et al. 2019a) that achieves privacy amplification by iteration. In Section 5 we outline how we resolve the technical difficulties to achieve privacy amplification for ADMM.

Materials in Full Version. In the full verison, we give the details for the general convex case and the strongly convex case. Moreover, we apply the techniques in (Feldman et al. 2018) to extend the privacy guarantees to all users. We also give the trade-off between privacy and utility. We perform a numerical illustration of our algorithms on a general Lasso problem. We have empirically confirmed that, as predicted theoretically, both the contraction factor and noise variance indeed affect the algorithm's convergence rates. We discuss potential improvements of parameters in our bounds.

2 Preliminaries

ADMM Convex Program. Suppose we have convex functions $\mathfrak{f} : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^\ell \to \mathbb{R}$, and linear transformations (also viewed as matrices) $A : \mathbb{R}^n \to \mathbb{R}^m$ and $B : \mathbb{R}^\ell \to \mathbb{R}^m$, and a vector $c \in \mathbb{R}^m$. The method ADMM is designed to tackle convex programs of the form:

$$\min_{x,y} \quad \mathfrak{f}(x) + g(y) \tag{1a}$$

$$\text{s.t.} \quad Ax + By = c \in \mathbb{R}^m \tag{1b}$$

$$x \in \mathbb{R}^n, y \in \mathbb{R}^\ell \tag{1c}$$

Function $\mathfrak{f}$ as an expectation functions. The function $\mathfrak{f}$ is derived from a distribution $\mathcal{D}$ of functions $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathfrak{f}(x) = \mathbf{E}_{f \leftarrow \mathcal{D}}[f(x)]$. We assume that the functions $f \in \mathcal{D}$ are differentiable. In the basic version, we assume that the algorithm has oracle access to the gradient $\nabla\mathfrak{f}(\cdot)$. However, in the stochastic version, there are T i.i.d. samples $(f_1, f_2, \ldots, f_T)$ from $\mathcal{D}$ and the algorithm only has oracle access to the gradient $\nabla f_t(\cdot)$ for each $t \in [T]$.

Notation. We use $\langle\cdot,\cdot\rangle$ to represent the standard inner product operation and $\|x\| := \sqrt{\langle x, x\rangle}$ to mean the usual Euclidean norm. We use $\mathbb{I}$ to denote the identity map (or matrix) in the appropriate space. Since a linear transformation A can be interpreted as a matrix multiplication, we use the transpose notation $A^\top : \mathbb{R}^m \to \mathbb{R}^n$ to denote the adjoint of A. We also consider the *operator norm* $\|A\| := \sup_{x \neq y} \frac{\|Ax - Ay\|}{\|x - y\|}$.

Smoothness Assumption. We assume that every function f in the support of $\mathcal{D}$ is differentiable and L-smooth for some $L > 0$; in other words, $\nabla f(\cdot)$ is L-Lipschitz, i.e., for all x and x', we have $\|\nabla f(x) - \nabla f(x')\| \leq L\|x - x'\|$. (To avoid too many parameters, later on we will mostly use $\eta = \frac{1}{L}$ in the algorithm description.)

Algorithm 1: One ADMM Iteration.

Input: Previous $(x_t, \lambda_t) \in \mathbb{R}^n \times \mathbb{R}^m$ and function $f_{t+1} : \mathbb{R}^n \to \mathbb{R}$.
Output: (x_{t+1}, λ_{t+1})

```
$y_t \leftarrow \mathcal{G}(\lambda_t - \beta A x_t)$        //pick canonical
    minimizer (see Lemma 2.1)
$\lambda_{t+1} \leftarrow \lambda_t - \beta(Ax_t + By_t - c)$
$x_{t+1} \leftarrow \mathcal{F}^{\nabla f_{t+1}}(x_t, y_t, \lambda_{t+1})$ //oracle access
    to $\nabla f_{t+1}(\cdot)$ (see Lemma 2.2)
return $(x_{t+1}, \lambda_{t+1})$  //$y_{t+1} \leftarrow \mathcal{G}(\lambda_{t+1} - \beta A x_{t+1})$
    can be recovered from $(x_{t+1}, \lambda_{t+1})$
```

Augmented Lagrangian Function. Recall that we have primal variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^\ell$, and the dual variable $\lambda \in \mathbb{R}^m$ corresponds to the feasibility constraint (1b). For some parameter $\beta > 0$, the following augmented Lagrangian function is considered in the literature:

$$\mathfrak{f}(x) + g(y) - \langle\lambda, Ax + By - c\rangle + \frac{\beta}{2}\|Ax + By - c\|^2.$$

The parameter $\beta > 0$ is chosen to offer a tradeoff between the approximations of the original objective function $\mathfrak{f}(x) + g(y)$ versus the feasibility constraint $Ax + By = c$, where a larger value of β means that more importance is placed on the feasibility constraint.

Augmented Lagrangian Function with First Order Approximation for Differentiable f. The first order approximation of f (with respect to some current $\widehat{x}$) is considered in the literature (Ouyang et al. 2015; Li and Lin 2019) as follows. As we shall see in Algorithm 1, each iteration just needs one gradient oracle access for $\nabla f(\cdot)$. Unless otherwise stated, we consider this *gradient* variant of ADMM throughout the paper.

$$\mathcal{L}^f_{\widehat{x}}(x, y, \lambda) := f(\widehat{x}) + \langle\nabla f(\widehat{x}), x - \widehat{x}\rangle + \mathcal{H}(x, y, \lambda) + \frac{1}{2\eta}\|x - \widehat{x}\|^2 \tag{2a}$$

$$\mathcal{H}(x, y, \lambda) := g(y) - \langle\lambda, Ax + By - c\rangle + \frac{\beta}{2}\|Ax + By - c\|^2 \tag{2b}$$

One ADMM Iteration. On a high level, ADMM is an iterative method. The three variables are updated in a round-robin fashion as described in Algorithm 1.

Given (x_{t+1}, λ_{t+1}), we can recover $y_{t+1} \leftarrow \mathcal{G}(\lambda_{t+1} - \beta A x_{t+1})$ deterministically as in Lemma 2.1. Hence, we only need to pass variables in the (x, λ)-space between consecutive iterations and treat y as an intermediate variable within each iteration.

Lemma 2.1 (Local Optimization for g)**.** *There exists $\mathcal{G} : \mathbb{R}^m \to \mathbb{R}^\ell$ such that for all x and λ, $\mathcal{G}(\lambda - \beta Ax) = \arg\min_y \mathcal{H}(x, y, \lambda)$.*[4]

Moreover, for any $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$, $y_1 = \mathcal{G}(\lambda - \beta Ax)$ and $y \in \mathbb{R}^\ell$, we have

$$g(y_1) - g(y) \leq \langle\lambda - \beta(Ax + By_1 - c), B(y_1 - y)\rangle.$$

[4] Note that the minimizer might not be unique. In practice, some deterministic method can pick a canonical value, or alternatively, we can invoke the Axiom of Choice such that $\mathcal{G}$ returns only one value in $\mathbb{R}^\ell$.

Proof. Observe that we can express $\mathcal{H}(x, y, \lambda) = \varphi(y) - \langle \lambda - \beta Ax, y\rangle + \vartheta(x, \lambda)$ for some functions $\varphi(y)$ and $\vartheta(x, \lambda)$. Observe that the variables (x, λ) and y only interact in the middle inner product term. Therefore, fixing (x, λ), $\arg\min_y \mathcal{H}(x, y, \lambda)$ is a function of $\lambda - \beta Ax$.

The optimality of y_1 implies that $0 \in \partial_y \mathcal{H}(x, y_1, \lambda) = \partial g(y_1) - B^\top \lambda + \beta B^\top(Ax + By_1 - c)$.

Hence, we have $s_1 := B^\top(\lambda - \beta(Ax + By_1 - c)) \in \partial g(y_1)$. The convexity of g implies that for all $y \in \mathbb{R}^\ell$, $g(y_1) - g(y) \leq \langle s_1, y_1 - y\rangle = \langle \lambda - \beta(Ax + By_1 - c), B(y_1 - y)\rangle$. □

Lemma 2.2 (Local Optimization for f)**.** *Given differentiable and convex $f : \mathbb{R}^n \to \mathbb{R}$, define $\mathcal{F}^{\nabla f} : \mathbb{R}^n \times \mathbb{R}^\ell \times \mathbb{R}^m \to \mathbb{R}^n$ by $\mathcal{F}^{\nabla f}(\widehat{x}, y, \lambda) := (\mathbb{I} + \eta\beta A^\top A)^{-1}\{\widehat{x} - \eta \cdot [\nabla f(\widehat{x}) + A^\top(\beta(By - c) - \lambda)]\}$. Then, it follows that $\mathcal{F}^{\nabla f}(\widehat{x}, y, \lambda)$ is the unique optimizer for problem $\min_x \mathcal{L}^f_{\widehat{x}}(x, y, \lambda)$, i.e. $\mathcal{F}^{\nabla f}(\widehat{x}, y, \lambda) = \arg\min_x \mathcal{L}^f_{\widehat{x}}(x, y, \lambda)$.*

Moreover, if $x_1 = \mathcal{F}^{\nabla f}(\widehat{x}, y, \lambda)$, then we have:
$\nabla f(\widehat{x}) = A^\top(\lambda - \beta(Ax_1 + By - c)) + \frac{1}{\eta} \cdot (\widehat{x} - x_1)$.

Proof. One can check that $\nabla_x \mathcal{L}^f_{\widehat{x}}(x, y, \lambda) = \nabla f(\widehat{x}) - A^\top \lambda + \beta A^\top(Ax + By - c) + \frac{1}{\eta} \cdot (x - \widehat{x})$.

Setting $\nabla_x \mathcal{L}^f_{\widehat{x}}(x_1, y, \lambda) = 0$ and observing that $(\mathbb{I} + \eta\beta A^\top A)$ has only eigenvalues at least 1 give the result. □

Private vs Public Information. We consider the scenario with T users, where each user $t \in [T]$ samples some function f_t from $\mathcal{D}$ independently. Each user t considers its function f_t as private information and as seen in Algorithm 1, when a user participates in one each iteration of ADMM, only oracle access to $\nabla f_t(\cdot)$ is sufficient, but the resulting information may leak private information about f_t. On the other hand, all other objects such as g, A, B, c and the initialization (x_0, λ_0) are considered public information.

Randomness in Privacy Model. In this paper, our privacy analysis does not exploit the randomness involved in sampling the functions, but is used in the analysis of convergence rates in the full version. Instead, for privacy analysis, we may assume that the (private) sequence of functions $(f_t : t \in [T])$ is fixed, which also determines that in iteration $t \in [T]$ of ADMM, the function f_t from user t will be accessed (via the gradient oracle).

Definition 2.3 (Neighboring Functions)**.** For $\Delta \geq 0$, define a (symmetric) neighboring relation $\sim_\Delta$ on functions in $\mathcal{D}$ such that two functions $f \sim_\Delta f'$ are neighboring if for all $x \in \mathbb{R}^n$, $\|\nabla f(x) - \nabla f'(x)\| \leq \Delta$.

Noisy ADMM for Local Privacy. *Where should noised be added?* In iteration $t + 1 \in [T]$ for Algorithm 1, the private information f_{t+1} of user $t + 1$ is accessed only in the computation of x_{t+1} in line 3 via the gradient oracle $\nabla f_{t+1}(\cdot)$.

Informal definition of local privacy. Suppose some (x_t, λ_t) is returned at the end of the iteration t. Consider two neighboring functions $f_{t+1} \sim_\Delta f'_{t+1}$, which are used in two scenarios of executing iteration $t + 1$ with the same input (x_t, λ_t). Local privacy for user $t + 1$ means that as long as the two functions f_{t+1} and f'_{t+1} are neighboring, the corresponding (random) $\widetilde{x}_{t+1}$ and $\widetilde{x}'_{t+1}$ from the two scenarios will have close distributions that can be quantified using *divergence* (see Section 3).

A standard way to achieve local privacy (with respect to the neighboring notion defined above) for user $t + 1$ is to sample some noise $N_{t+1} \in \mathbb{R}^n$, e.g., Gaussian noise $\mathcal{N}(0, \sigma^2\mathbb{I})$ with some appropriate variance σ^2. Then, the generated noise N_{t+1} is used to return the masked value $\widetilde{x}_{t+1} \leftarrow x_{t+1} + N_{t+1}$. In this paper, we will refer to this as the *noisy variant* of Algorithm 1, or *noisy ADMM*.

Observe that as far as local privacy is concerned, there is no need to mask y_t or λ_{t+1}, whose computation does not involve the private function f_{t+1}.

Privacy Amplification by Iteration. Observe that in each iteration $t \in [T]$, some noise N_t is sampled to mask the value x_t to achieve local privacy for user t. Privacy amplification by iteration refers to the privacy analysis from the perspective of the **first user** (i.e. $t = 1$). Since there is so much randomness generated in all T iterations, will the privacy guarantee for the finally returned $(\widetilde{x}_T, y_T, \lambda_T)$ be amplified with respect to the first user? The challenge here is that in each iteration t, noise is added only to the computation of the x variable, but not to the y and λ variables.

Finally, to achieve privacy amplification for ADMM, we need to transform the problem instance into an appropriate form, whose significance will be apparent in Section 5.2.

Remark 2.4 (Transformation of the Linear Constraints)**.** By Gaussian elimination, we may assume without loss of generality that $m \leq n$ and the matrix A has the form $A = [\mathbb{I}_m \mid D]$ for some $m \times (n - m)$ matrix D. However, after the process of Gaussian elimination, we may have extra linear constraints of the form $\widehat{B}y = \widehat{c}$, which can be absorbed into a modified convex function $\widehat{g}$ in the following:

$$\widehat{g}(y) := \begin{cases} g(y), & \text{if } \widehat{B}y = \widehat{c}; \\ +\infty, & \text{otherwise.} \end{cases}$$

Observe this transformation does not change n and ℓ. However, if initially m is strictly greater than n, then the transformation ensures that $m \leq n$ afterwards.

3 Rényi and Zero-Concentrated Differential Privacy Background

We present the most essential ideas related to differential privacy (Dwork 2006), Rényi (Mironov 2017) and zero-concentrated (Bun and Steinke 2016) differential privacy here.

Recall that the differential privacy states that for neighboring inputs, the two corresponding outputs have distributions that are close with respect to some notion of divergence.

Definition 3.1 (Rényi Divergence (Rényi et al. 1961))**.** Given distributions P and Q over some sample space $\mathcal{O}$, the Rényi divergence of order $\alpha > 1$ between them is:

$$\mathsf{D}_\alpha(P\|Q) := \frac{1}{\alpha-1}\ln \mathbf{E}_{x\leftarrow Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha.$$

Zero-Concentrated Divergence. To define zero-concentrated differential privacy (zCDP), we consider the following divergence:

$\mathsf{D}^{\mathsf{z}}(P\|Q) := \sup_{\alpha>1} \frac{1}{\alpha} \cdot \mathsf{D}_\alpha(P\|Q)$.

Fact 3.2 (Rényi and D^{z}-Divergence between Gaussian Noises (Bun and Steinke 2016))**.** *For any vectors $x, x' \in \mathbb{R}^n$, we have:*

- $\mathsf{D}_\alpha(\mathcal{N}(x,\sigma^2\mathbb{I})\|\mathcal{N}(x',\sigma^2\mathbb{I})) = \frac{\alpha\|x-x'\|^2}{2\sigma^2}$*, for any* $\alpha \geq 1$.
- $\mathsf{D}^{\mathsf{z}}(\mathcal{N}(x,\sigma^2\mathbb{I}_n)\|\mathcal{N}(x',\sigma^2\mathbb{I}_n)) = \frac{\|x-x'\|^2}{2\sigma^2}$.

The following fact states the properties of D^{z}-divergence that we need.

Fact 3.3 (Properties for D^{z}-Divergence (Bun and Steinke 2016, Lemma 2.2)[5])**.** *Suppose (X, Y) and (X', Y') are joint distributions with the same support. Then, we have the following conclusions.*

(a) Data processing inequality. *It holds that* $\mathsf{D}^{\mathsf{z}}(Y\|Y') \leq \mathsf{D}^{\mathsf{z}}((X,Y)\|(X',Y'))$.

(b) Uniform to Average Bounds (a.k.a. Quasi-convexity). *Suppose there exists $\epsilon_2 \geq 0$ such that for any x in the support of X and X', the conditional distributions satisfy:*
$\mathsf{D}^{\mathsf{z}}((Y|X=x)\|(Y'|X'=x)) \leq \epsilon_2$.
Then, the marginal distributions satisfy $\mathsf{D}^{\mathsf{z}}(Y\|Y') \leq \epsilon_2$.

(c) Adaptive composition. *Suppose, in addition to the condition in (b), there exists $\epsilon_1 \geq 0$ such that* $\mathsf{D}^{\mathsf{z}}(X\|X') \leq \epsilon_1$.
Then, it holds that $\mathsf{D}^{\mathsf{z}}((X,Y)\|(X',Y')) \leq \epsilon_1 + \epsilon_2$.

4 Summary of Coupling Framework to Achieve Privacy Amplification by Iteration

We give a short summary of the coupling framework in (Balle et al. 2019a) used to analyze privacy amplification in an iterative process.

Iteration interpreted as a Markov operator. Suppose the information passed between consecutive iterations is an element in some sample space Ω (which is also equipped with some norm[6] $\|\cdot\|$). Since each iteration uses fresh randomness, it can be represented as a Markov operator $K : \Omega \to \mathcal{P}(\Omega)$. If $z \in \Omega$ is the input to an iteration, then $K(z)$ represents the distribution of the output returned by that iteration. We use $\mathcal{K}$ to denote the collection of *iteration* Markov operators, each of which corresponds to some iteration. For each $K \in \mathcal{K}$ and $z \in \Omega$, we also view that the process samples some fresh randomness N (independent of z) from an appropriate distribution, and the output is a deterministic function of (z, N). (We will use this randomness N when we consider the notion of *coupling* described below.)

Notation. Observe that we can naturally extend a Markov operator to $K : \mathcal{P}(\Omega) \to \mathcal{P}(\Omega)$. If $\mu \in \mathcal{P}(\Omega)$ is a distribution, then $K(\mu) \in \Omega$ is the distribution corresponding to the following sampling process: (i) first, sample $z \in \Omega$ from μ, (ii) second, return a sample from the distribution $K(z)$.

Differing slightly from the notation in (Balle et al. 2019a), we denote the composition of Markov operators using the function notation, i.e., $(K_2 \circ K_1)(\mu) = K_2(K_1(\mu))$.

Coupling. Given two distributions $\mu, \nu \in \mathcal{P}(\Omega)$, a coupling π from μ to ν is a joint distribution on $\Omega \times \Omega$ such that marginal distributions for the two components are μ and ν, respectively.

When the same iterator operator $K \in \mathcal{K}$ is applied in two different scenarios, the *natural coupling* refers to using the same aforementioned randomness N sampled within the iteration process for both scenarios.

The following notion gives a uniform bound on the distance between two distributions.

Definition 4.1 (∞-Wasserstein Distance)**.** Given distributions μ and ν on some normed space Ω and $\Delta \geq 0$, a coupling π from μ to ν is a witness that the (infinity) Wasserstein distance $\mathsf{W}(\mu,\nu) \leq \Delta$ if for all $(z,z') \in \mathsf{supp}(\pi)$, $\|z - z'\| \leq \Delta$.

The distance $\mathsf{W}(\mu,\nu)$ is the infimum of the collection of Δ for which such a witness π exists.

The following theorem paraphrases the privacy amplification result in (Balle et al. 2019a, Theorem 4) for the special case of non-expansion.

Theorem 4.2 (Privacy Amplification by Iteration (Balle et al. 2019a))**.** *Suppose the collection $\mathcal{K}$ of iteration operators satisfies the following conditions.*

(A) Non-Expansion. *There is some $0 < L \leq 1$ such that for any $K \in \mathcal{K}$ and $z, z' \in \Omega$, there exists a witness (e.g., the natural coupling) for* $\mathsf{W}(K(z), K(z')) \leq L\|z - z'\|$.

(B) One-step Privacy. *There exists a constant $C_{\mathcal{K}} > 0$ (depending on $\mathcal{K}$) such that for any $K \in \mathcal{K}$ and $z, z' \in \Omega$, it holds that:* $\mathsf{D}^{\mathsf{z}}(K(z)\|K(z')) \leq C_{\mathcal{K}} \cdot \|z - z'\|^2$.

Then, given for any $T \geq 1$ iterator operators $K_1, K_2, \ldots, K_T \in \mathcal{K}$ and $z, z' \in \Omega$, it holds that

$\mathsf{D}^{\mathsf{z}}((K_T \circ \cdots \circ K_1)(z)\|(K_T \circ \cdots \circ K_1)(z')) \leq \frac{C_{\mathcal{K}} L^{T-1}}{T} \cdot \|z - z'\|^2$.

4.1 Technical Challenges for Applying the Framework to ADMM

We outline how we resolve the challenges when we apply this coupling framework to our gradient variant of ADMM.

Non-expansive iteration. It is crucial in (Feldman et al. 2018; Balle et al. 2019a) that before adding noise, each iteration corresponds to a non-expansive mapping acting on the variable space, which is also true for the proximal variant considered in (Cyffers, Bellet, and Basu 2023). As aforementioned, for the variant of ADMM that uses gradient oracle, one iteration (in Algorithm 1) corresponds to a transition in the (x, λ)-space. However, it can be shown (in the full version) that this transition may correspond to a strictly expanding mapping under the usual norm.

Our idea is to design a customized norm in the (x, λ)-space that (i) is suitable for analyzing the privacy of ADMM and (ii) satisfies the condition that one ADMM iteration corresponds to a non-expansive (or strictly contractive for

[5] The original results (Bun and Steinke 2016, Lemma 2.2) have been stated in terms of the Rényi divergence D_α, but they have also shown that they can be readily generalized to D^{z}.

[6] We actually just need the linearity property $\|az\|^2 = a^2\|z\|^2$ for $a \in \mathbb{R}$ and $z \in \Omega$.

strongly convex objective functions) mapping under this customized norm. Our proof technique is reminiscent of the convergence proofs for ADMM, which crucially utilize the optimality conditions for variable updates and the (strong) convexity of the objective functions.

One-step privacy. In (Feldman et al. 2018; Balle et al. 2019a), the variable produced in each iteration is totally masked at every coordinate with Gaussian noise before passing to the next iteration. Hence, it is somehow straightforward (also using the aforementioned non-expansive property) to achieve some privacy guarantee for one iteration in terms of D^{z}-divergence. Indeed, this can be readily applied to the proximal variant of ADMM because the noise added to the x variable is transformed by a linear mapping to to mask the λ variable, which is the only variable passed to the next iteration.

However, the case for passing both variables x and λ is more complicated. As aforementioned, in each iteration, the sampled noise is used to mask only the x variable. This means that for the variable in the (x, λ)-space returned in one iteration, the λ-component receives no noise and is totally exposed. Therefore, no matter how much noise is used to mask x variable, the resulting privacy analysis for one iteration will still give a D^{z}-divergence of $+\infty$.

Our innovative idea is to consider one step as consisting of two noisy ADMM iterations. The very informal intuition is that the two copies of independent noises from two iterations can each be used to mask one component of (x, λ) in the result at the end of the two iterations. Specifically, one copy of the noise is used to mask the variable λ (via a linear transformation) and the other copy is used to mask the variable x. To avoid complicated integral calculations, we perform the relevant D^{z}-divergence analysis using the tools of adaptive composition of private mechanisms.

5 Summary of Techniques for Amplification by Iteration for ADMM

We give the technical details for applying the coupling framework described in Section 4 to achieve privacy amplification for ADMM.

Recall that the high level goal is to amplify the privacy guarantee for the user in the **first** iteration via the randomness in subsequent iterations. We will **not** exploit the masking randomness (for the variable x) generated in the first iteration in the privacy amplification analysis. Therefore, for the purpose of privacy amplification, the starting points for the two scenarios are two inputs (x_0, λ_0) and (x'_0, λ_0), where $x_0 \neq x'_0$ and the λ components are the same. Below is the main technical result.

Theorem 5.1 (Privacy Amplification by Iteration for ADMM)**.** *Suppose given two input scenarios (x_0, λ_0) and (x'_0, λ_0), a total of $2T$* noisy *ADMM iterations in Algorithm 1 are applied to each input scenario, where for each iteration $t \in [2T]$, the same function f_t is used in both scenarios and fresh randomness N_t drawn from Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathbb{I}_n)$ is used to produce masked $\widetilde{x}_t \leftarrow x_t + N_t$ (that is passed to the next iteration together with λ_t). Then, the corresponding output distributions from the two scenarios satisfy:*

$$\mathsf{D}^{\mathsf{z}}((\widetilde{x}_{2T}, \lambda_{2T}) \| (\widetilde{x}'_{2T}, \lambda'_{2T})) \leq \frac{C}{T} \cdot \|x_0 - x'_0\|^2,$$

where $C := \frac{1}{2\sigma^2} \max\{2, \frac{3}{\beta\eta}\} \cdot (1 + \beta\eta \cdot \|A\|^2)$.

Remark. Observe that in Theorem 5.1, the parameter C has a dependency of $O(\frac{1}{\beta\eta})$ on β and η. We will discuss potential improvements in the full version.

As described in Section 4.1, the main technical challenges are how to achieve (A) non-expansion (in Section 5.1) and (B) one-step privacy (in Section 5.2). After achieving those two key properties, we will show how everything fits together to achieve Theorem 5.1 in Section 5.3.

5.1 Achieving Non-expansion via Customized Norm

As discussed in Section 4.1, one ADMM iteration as in Algorithm 1 may produce a strictly expanding mapping under the usual norm. We consider the following specialized norm.

Definition 5.2 (Customized Norm)**.** Using the ADMM parameters η and β from Section 2, we define a customized norm. For $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$, $\|(x, \lambda)\|_*^2 := \|x\|^2 + \frac{\eta}{\beta} \cdot \|\lambda - \beta Ax\|^2$.

Remark. Even though we use the term "norm", we only need the linearity property, i.e., for all $a \in \mathbb{R}$, $\|(ax, a\lambda)\|_*^2 = a^2 \cdot \|(x, \lambda)\|_*^2$. Note that we do not need any triangle inequality for the customized norm.

Lemma 5.3 (ADMM Iteration is Non-expansive with Customized Norm)**.** *Suppose one ADMM iteration in Algorithm 1 is applied to two different inputs (x_t, λ_t) and (x'_t, λ'_t) with the same function f that is convex and $\frac{1}{\eta}$-smooth. Then, the corresponding two outputs (x_{t+1}, λ_{t+1}) and $(x'_{t+1}, \lambda'_{t+1})$ satisfy: $\|(x_{t+1} - x'_{t+1}, \lambda_{t+1} - \lambda'_{t+1})\|_*^2 \leq \|(x_t - x'_t, \lambda_t - \lambda'_t)\|_*^2$.*

The detailed proof is given in the full version. Even though the proof is technical, it uses the same intuition as an ADMM convergence proof (He and Yuan 2012). Specifically, standard inequalities related to optimality conditions for updating the x and y variables are used.

5.2 Achieving One-Step Privacy

As described in Section 4.1, the randomness in one noisy ADMM iteration is not sufficient to achieve one-step privacy (condition (B) in Theorem 4.2) because only the x component of the output of Algorithm 1 is masked with noise, while the λ component is totally exposed.

Incorporating two noisy ADMM iterations into a single Markov Operator. Our novel idea is to let each Markov operator represent two ADMM iterations. For instance, an operator $K \in \mathcal{K}$ in the collection corresponds to iterations $t+1$ and $t+2$, which use two $\frac{1}{\eta}$-smooth convex functions f_{t+1} and f_{t+2}, respectively. Given some input $(\widetilde{x}_t, \lambda_t)$, the application $K(\widetilde{x}_t, \lambda_t)$ of the operator K is the randomized process for executing two noisy ADMM interations, whose source of randomness is two independent copies N_{t+1} and N_{t+2} (used for masking each x variable) of Gaussian noise with some appropriate variance σ^2.

Algorithm 2: Mechanism $\mathcal{M}_1$

```
Input: $(\widetilde{x}_t, \lambda_t)$ and fixing the last $n-m$ coordinates of $N_{t+1}$ to be $\mathfrak{z}$.
Output: $\widetilde{w}_{t+1}$
$y_t \leftarrow \mathcal{G}(\lambda_t - \beta A\widetilde{x}_t)$
$\lambda_{t+1} \leftarrow \lambda_t - \beta(A\widetilde{x}_t + By_t - c)$
$x_{t+1} \leftarrow \mathcal{F}^{f_{t+1}}(\widetilde{x}_t, y_t, \lambda_{t+1})$   //first 3 lines same as Algorithm 1
$w_{t+1} \leftarrow \lambda_{t+1} - \beta Ax_{t+1}$
Sample fresh $U_{t+1}$ from $\mathcal{N}(0, \sigma^2\mathbb{I}_m)$.
return $\widetilde{w}_{t+1} \leftarrow w_{t+1} - \beta D\mathfrak{z} - \beta U_{t+1}$
```

Proof Setup. Given two input scenarios $(\widetilde{x}_t, \lambda_t)$ and $(\widetilde{x}'_t, \lambda'_t)$, our goal is to derive an upperbound for the divergence $\mathsf{D}^{\mathsf{z}}(K(\widetilde{x}_t, \lambda_t) \| K(\widetilde{x}'_t, \lambda'_t))$ that is defined in Section 3.

Recall that in Remark 2.4, we have transformed the problem by Gaussian elimination such that $A = [\mathbb{I}_m D]$ for some $m \times (n-m)$ matrix D. In our privacy analysis, we do not actually need to use the randomness of all n coordinates of N_{t+1} (but we still need all n coordinates of N_{t+2}). We use $U_{t+1} \sim \mathcal{N}(0, \sigma^2\mathbb{I}_m)$ to represent the first m coordinates of N_{t+1}. In both scenarios, we will fix the last $n-m$ coordinates of N_{t+1} and denote this common part as $\mathfrak{z} \in \mathbb{R}^{n-m}$ (which is no longer random). By Fact 3.3(b), any uniform upperbound on the D^{z}-divergence after conditioning on $\mathfrak{z}$ will also be an upperbound for the original divergence. Observe that $AN_{t+1} = U_{t+1} + D\mathfrak{z}$.

Expressing Markov Operator K as an Adaptive Composition of Two Private Mechanisms. Instead of directly working with probability density function of $(\widetilde{x}_{t+2}, \lambda_{t+2})$ in the analysis of D^{z}-divergence, we will use properties of D^{z}-divergence in Fact 3.3, whose proofs in the literature have already incorporated the technical manipulation of integrals. Our approach is to analyze the divergence in the language of adaptive composition of private mechanisms with which most readers have some familiarity.

How to decompose $(\widetilde{x}_{t+2}, \lambda_{t+2})$? No matter whether one wants to directly analyze the probability density functions or make use of adaptive composition, one technical hurdle is that given one component of the pair $(\widetilde{x}_{t+2}, \lambda_{t+2})$, the conditional distribution of the other component is not easy to analyze. For the composition, it actually suffices to consider an intermediate variable $w_{t+1} = \lambda_{t+1} - \beta Ax_{t+1}$ and its masked variant $\widetilde{w}_{t+1} = \lambda_{t+1} - \beta A(x_{t+1} + N_{t+1})$. We shall see in Algorithm 3 that λ_{t+2} is a deterministic function of $\widetilde{w}_{t+1}$. Hence, by the data processing inequality in Fact 3.3(a), it suffices to analyze a (randomized) composition that takes input $(\widetilde{x}_t, \lambda_t)$ and returns the pair $(\widetilde{w}_{t+1}, \widetilde{x}_{t+2})$.

Divergence Analysis. When we consider two inputs $(\widetilde{x}_t, \lambda_t)$ and $(\widetilde{x}'_t, \lambda'_t)$, we use a superscript to indicate variables associated with the second input. The detailed analysis is given in the full version. It is shown that the adaptive composition $\mathcal{M}_2 \circ \mathcal{M}_1$ is equivalent to the operator K, and a privacy composition proof leads to the following result.

Algorithm 3: Mechanism $\mathcal{M}_2$

```
Input: $(\widetilde{x}_t, \lambda_t, \widetilde{w}_{t+1})$ and fixing the last $n-m$ coordinates of $N_{t+1}$ to be $\mathfrak{z}$.
Output: $\widetilde{x}_{t+2}$
$y_t \leftarrow \mathcal{G}(\lambda_t - \beta A\widetilde{x}_t)$
$\lambda_{t+1} \leftarrow \lambda_t - \beta(A\widetilde{x}_t + By_t - c)$
$x_{t+1} \leftarrow \mathcal{F}^{f_{t+1}}(\widetilde{x}_t, y_t, \lambda_{t+1})$
$w_{t+1} \leftarrow \lambda_{t+1} - \beta Ax_{t+1}$   //first 4 lines the same as Algorithm 2
$U_{t+1} \leftarrow \frac{1}{\beta}(w_{t+1} - \widetilde{w}_{t+1}) - D\mathfrak{z}$   //randomness ``reconstruction''; $A = [\mathbb{I}_m\, D]$
$\widetilde{x}_{t+1} \leftarrow x_{t+1} + (U_{t+1}, \mathfrak{z})$   //$N_{t+1} = (U_{t+1}, \mathfrak{z})$; $\widetilde{w}_{t+1} = \lambda_{t+1} - \beta A\widetilde{x}_{t+1}$
$y_{t+1} \leftarrow \mathcal{G}(\widetilde{w}_{t+1})$ //2nd iteration of ADMM Algorithm 1 with input $(\widetilde{x}_{t+1}, \lambda_{t+1})$
$\lambda_{t+2} \leftarrow \widetilde{w}_{t+1} - \beta(By_{t+1} - c)$   //$\lambda_{t+2}$ is a deterministic function of $\widetilde{w}_{t+1}$
$x_{t+2} \leftarrow \mathcal{F}^{f_{t+2}}(\widetilde{x}_{t+1}, y_{t+1}, \lambda_{t+2})$   //oracle access to $\nabla f_{t+2}(\cdot)$
Sample fresh $N_{t+2}$ from $\mathcal{N}(0, \sigma^2\mathbb{I}_n)$.
return $\widetilde{x}_{t+2} \leftarrow x_{t+2} + N_{t+2}$
```

Lemma 5.4 (One-Step Privacy of Operator K)**.** *Given inputs $(\widetilde{x}_t, \lambda_t)$ and $(\widetilde{x}'_t, \lambda'_t)$ and Markov operator $K \in \mathcal{K}$ (corresponding to two ADMM iterations using convex $\frac{1}{\eta}$-smooth functions and $\mathcal{N}(0, \sigma^2\mathbb{I}_n)$ noise), we have:*

$$\mathsf{D}^{\mathsf{z}}\left(K(\widetilde{x}_t, \lambda_t) \middle\| K(\widetilde{x}'_t, \lambda'_t)\right) \leq C_{\mathcal{K}} \left\|(\widetilde{x}_t - \widetilde{x}'_t, \lambda_t - \lambda'_t)\right\|_*^2,$$

where $C_{\mathcal{K}} = \frac{1}{2\sigma^2} \max\left(2, \frac{3}{\eta\beta}\right)$.

5.3 Combining Everything Together to Achieve Privacy Amplification by Iteration for ADMM

Proof of Theorem 5.1. Theorem 4.2 can be applied, because we have already established the corresponding condition (A) can be achieved by the result in Section 5.1 and condition (B) is attained in Section 5.2. The desired bound follows because $2T$ noisy ADMM iterations correspond to the composition of T Markov operators in the collection $\mathcal{K}$. The detailed proof is in the full version. □

6 Conclusion

We have applied the coupling framework (Balle et al. 2019a) to achieve privacy amplification by iteration for ADMM. Specifically, we have recovered the factor of $\frac{1}{T}$ (or $\frac{L^T}{T}$ for some $0 < L < 1$ in the strongly convex case) in the D^{z}-divergence as the number T of iteration increases. We have performed experiments to evaluate the empirical performance of our methods in the full version.

Acknowledgments

This research was partially supported by the Hong Kong RGC grants 17202121 and 17203122.

References

Balle, B.; Barthe, G.; and Gaboardi, M. 2018. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *NeurIPS*, 6280–6290.

Balle, B.; Barthe, G.; Gaboardi, M.; and Geumlek, J. 2019a. Privacy Amplification by Mixing and Diffusion Mechanisms. In *NeurIPS*, 13277–13287.

Balle, B.; Bell, J.; Gascón, A.; and Nissim, K. 2019b. The Privacy Blanket of the Shuffle Model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, 638–667. Springer.

Boyd, S. P.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1): 1–122.

Bun, M.; and Steinke, T. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *TCC (B1)*, volume 9985 of *Lecture Notes in Computer Science*, 635–658.

Chan, T.-H. H.; Xie, H.; and Zhao, M. 2023. Privacy Amplification by Iteration for ADMM with (Strongly) Convex Objective Functions. arXiv:2312.08685.

Chaudhuri, K.; and Mishra, N. 2006. When Random Sampling Preserves Privacy. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, 198–213. Springer.

Cheu, A.; Smith, A. D.; Ullman, J. R.; Zeber, D.; and Zhilyaev, M. 2019. Distributed Differential Privacy via Shuffling. In *EUROCRYPT (1)*, volume 11476 of *Lecture Notes in Computer Science*, 375–403. Springer.

Cyffers, E.; Bellet, A.; and Basu, D. 2023. From Noisy Fixed-Point Iterations to Private ADMM for Centralized and Federated Learning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 6683–6711. PMLR.

Dwork, C. 2006. Differential Privacy. In *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, 1–12. Springer.

Erlingsson, Ú.; Feldman, V.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Thakurta, A. 2019. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *SODA*, 2468–2479. SIAM.

Feldman, V.; Mironov, I.; Talwar, K.; and Thakurta, A. 2018. Privacy Amplification by Iteration. In *FOCS*, 521–532. IEEE Computer Society.

Gabay, D.; and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics With Applications*, 2: 17–40.

He, B.; and Yuan, X. 2012. On the O(1/n) Convergence Rate of the Douglas-Rachford Alternating Direction Method. *SIAM J. Numer. Anal.*, 50(2): 700–709.

Li, H.; and Lin, Z. 2019. Accelerated Alternating Direction Method of Multipliers: An Optimal O(1 / K) Nonergodic Analysis. *J. Sci. Comput.*, 79(2): 671–699.

Mironov, I. 2017. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, 263–275. IEEE.

Ouyang, H.; He, N.; Tran, L. Q.; and Gray, A. G. 2013. Stochastic Alternating Direction Method of Multipliers. In *ICML (1)*, volume 28 of *JMLR Workshop and Conference Proceedings*, 80–88. JMLR.org.

Ouyang, Y.; Chen, Y.; Lan, G.; and Jr., E. P. 2015. An Accelerated Linearized Alternating Direction Method of Multipliers. *SIAM J. Imaging Sci.*, 8(1): 644–681.

Rényi, A.; et al. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California.

Shang, F.; Xu, T.; Liu, Y.; Liu, H.; Shen, L.; and Gong, M. 2021. Differentially Private ADMM Algorithms for Machine Learning. *IEEE Trans. Inf. Forensics Secur.*, 16: 4733–4745.

Zhang, T.; and Zhu, Q. 2016. A Dual Perturbation Approach for Differential Private ADMM-Based Distributed Empirical Risk Minimization. In *AISec@CCS*, 129–137. ACM.