# Gradient Descent with Linearly Correlated Noise: Theory and Applications to Differential Privacy

**Anastasia Koloskova**[*] EPFL, Switzerland
**Ryan McKenna** Google Research
**Zachary Charles** Google Research
**Keith Rush** Google Research
**Brendan McMahan** Google Research

## Abstract

We study gradient descent under linearly correlated noise. Our work is motivated by recent practical methods for optimization with differential privacy (DP), such as DP-FTRL, which achieve strong performance in settings where privacy amplification techniques are infeasible (such as in federated learning). These methods inject privacy noise through a matrix factorization mechanism, making the noise linearly correlated over iterations. We propose a simplified setting that distills key facets of these methods and isolates the impact of linearly correlated noise. We analyze the behavior of gradient descent in this setting, for both convex and non-convex functions. Our analysis is demonstrably tighter than prior work and recovers multiple important special cases exactly (including anti-correlated perturbed gradient descent). We use our results to develop new, effective matrix factorizations for differentially private optimization, and highlight the benefits of these factorizations theoretically and empirically.

## 1 Introduction

Differential privacy (DP) is a critical framework for designing algorithms with provable statistical privacy guarantees. DP stochastic gradient descent (DP-SGD, Abadi et al. [1]) is particularly important for enabling private empirical risk minimization (ERM) of machine learning models. Many works have analyzed the convergence behavior of DP ERM methods, including DP-SGD [5, 16, 48, 8]. However, obtaining good privacy/utility trade-offs with DP-SGD can require excessively large batch sizes or privacy amplification techniques such as subsampling [4, 5, 55] and shuffling [15, 16]. In some applications, including cross-device federated learning, limited and device-controlled client availability can make sampling or shuffling infeasible [21]. Even outside of such applications, many implementations of DP-SGD do not properly use the Poisson subsampling scheme analyzed by Abadi et al. [1] for amplification, and instead use a single fixed permutation of the dataset [7].

Kairouz et al. [20] propose an alternative method, DP-FTRL, which can attain good privacy/utility trade-offs without amplification. Their key insight is that for SGD-style algorithms, the variance on *prefix sums* $\mathbf{g}_0 + \cdots + \mathbf{g}_t$, $t \in \{1, \ldots, T\}$ of gradients $\mathbf{g}_j$ is more important than the variance on individual gradients. By adding carefully tailored noise that is *linearly correlated* over iterations to the gradients, one can reduce the error on the prefix sums, at the cost of increased error on the individual gradients, for a fixed privacy budget. The DP-FTRL mechanism is competitive with or better than DP-SGD, even without relying on privacy amplification, and enabled McMahan and Thakurta [31] to train the first differentially private machine learning model on user data in a production setting.

[*]Work performed while doing an internship at Google Research. Correspondence to: Anastasia Koloskova <anastasia.koloskova@epfl.ch>, Ryan McKenna <mckennar@google.com>.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

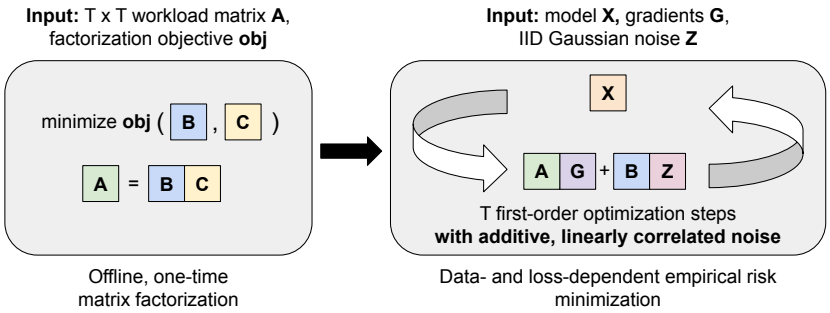



Figure 1: Two-stage MF-DP-FTRL workflow proposed by Denisov et al. [10]. The user selects a workload matrix $\mathbf{A}$ representing a desired first-order optimization method. Offline, the user finds a factorization $\mathbf{BC} = \mathbf{A}$, using an objective that balances ERM performance (as a function of $\mathbf{B}$) and privacy (as a function of $\mathbf{C}$). The user applies $\mathbf{A}$ to a downstream ERM task, but with linearly correlated additive noise governed by $\mathbf{B}$.

Denisov et al. [10], Choquette-Choo et al. [7] develop a refinement of DP-FTRL, MF-DP-FTRL, by formulating and solving an offline matrix factorization problem to find the "optimal" correlated noise structure under DP constraints. That is, for a fixed privacy level, they aim to find correlated noise structures that lead to improved optimization. A simplified diagram of their workflow is given in Fig. 1. However, (as we detail in Section 2) their offline factorization objective is based on an online convergence bound that is loose. This raises questions about whether there are factorization objectives that better capture convergence behavior of gradient descent algorithms with correlated noise.

In this paper we study this class of mechanisms more closely and provide a detailed analysis of linearly correlated noise from an optimization point of view. Our main contributions are as follows:

- We propose a novel stochastic optimization problem that extracts key facets of methods like (MF-)DP-FTRL, and which isolates the effects of linearly correlated noise on optimization.
- We derive convergence rates for gradient descent on smooth convex and non-convex functions in such settings that showcase the effect of linearly correlated noise and recover tight convergence rates in notable special cases. We use a novel proof technique that may be of independent interest.
- We use this theory to design a new objective for the offline matrix factorization workflow in Fig. 1. We show that solving this objective leads to MF-DP-FTRL mechanisms with improved convergence properties. We validate the mechanism empirically on a variety of datasets and tasks, matching or outperforming prior methods.

### 1.1 Related Work

**Matrix mechanisms for differential privacy.** Our work is closely related to differentially private optimization using matrix mechanisms [26]. Historically, such mechanisms were applied to linear statistical queries [25, 29, 14, 18]. Denisov et al. [10] and Choquette-Choo et al. [7] extended these mechanisms to the adaptive streaming setting, allowing their application to optimization with DP. Denisov et al. [10] show that this framework (MF-DP-FTRL) subsumes and improves the DP-FTRL algorithm [20]. Both DP-FTRL and MF-DP-FTRL improve privacy guarantees relative to DP-SGD [1] without amplification, and can be combined with techniques such as momentum for improved utility [46]. The aforementioned work focuses on methods for computing factorizations, privacy properties, and empirics. Our work studies the analytic relationship between the correlated noise induced by the MF-DP-FTRL framework and the downstream effect on optimization performance.

**SGD with correlated noise.** Stochastic noise in optimization arises in a variety of ways, including mini-batching [9] and explicit noise injection [11, 54, 19]. While most analyses of SGD assume this noise is independent across iterates, some work considers correlated noise. For example, shuffle SGD involves correlated noise due to sampling without replacement [33, 53]. Lucchi et al. [27]

use correlated Brownian motion to improve SGD's ability to explore the loss landscape. Recently, Orvieto et al. [37, 38] investigated anti-correlated noise as a way to impose regularization and improve generalization. We consider a linearly correlated noise model, and analyze its impact on SGD's convergence to critical points.

**SGD with biased noise.** Many algorithms can be viewed as SGD with structured but potentially biased noise, including SGD with (biased) compression [44, 17], delayed SGD [28, 12], local SGD [42], federated learning methods [22, 52, 34, 36], decentralized optimization methods [50, 23], and many others. Convergence analyses for such methods often use techniques like perturbed iterate analysis [28]. Correlated gradient noise also biases the gradient updates. However, as we show in Section 4, directly applying such techniques to linearly correlated noise does not lead to tight convergence guarantees.

## 2 Background

In this work, we focus on an empirical risk minimization (ERM) problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x},\xi_i)\right], \tag{1}$$

where $l(\mathbf{x},\xi_i)$ is the loss of a model $\mathbf{x}$ on a data point $\xi_i$, and $n$ is the training set size. We would like to solve (1) while guaranteeing some form of privacy for the training set. We focus on *differential privacy* (DP, [13]), a widely-used standard for anonymous data release. DP guarantees statistical difficulty in distinguishing whether or not a particular unit's data served as an input to a given algorithm, based on the algorithm's output. This protected unit may represent a single training example or a semantically higher-level unit like the entirety of a user's data.

While there are many methods for solving (1), we will follow Denisov et al. [10], Choquette-Choo et al. [7] and restrict to first-order algorithms $\mathfrak{A}$ that linearly combine (stochastic) gradients. Each algorithm $\mathcal{A}\in\mathfrak{A}$ is parameterized by a learning rate $\gamma>0$, a number of steps $T>0$, and scalars $\{a_{tj}\}_{1\le j\le t\le T}$. Given a starting point $\mathbf{x}_0$, $\mathcal{A}$ produces iterates $\mathbf{x}_t\in\mathbb{R}^d$ given by

$$\mathbf{x}_{t+1} = \mathbf{x}_0 - \gamma\mathcal{A}_t(\mathbf{g}_1,\dots,\mathbf{g}_t) \qquad\qquad \mathcal{A}_t(\mathbf{g}_1,\dots,\mathbf{g}_t) = \textstyle\sum_{j=1}^{t} a_{tj}\mathbf{g}_j$$

where $\mathbf{g}_t$ is a (mini-batch) gradient of $f$ computed at $\mathbf{x}_t$. This class encompasses a variety of first-order algorithms, including SGD [40], SGD with momentum [39, 35], and delayed SGD [2]. This class also captures algorithms that use learning rate scheduling, so long as the schedule is independent of the gradient values. We re-write the output of $\mathcal{A}$ in matrix notation by defining:

$$\mathbf{X} = [\mathbf{x}_1,\dots,\mathbf{x}_T]^\top\in\mathbb{R}^{T\times d},\ \mathbf{X}_0 = [\mathbf{x}_0,\dots,\mathbf{x}_0]^\top\in\mathbb{R}^{T\times d}$$
$$\mathbf{G} = [\mathbf{g}_1,\dots,\mathbf{g}_T]^\top\in\mathbb{R}^{T\times d},\ \mathbf{A} = [a_{ij}]_{1\le i,j\le T}\in\mathbb{R}^{T\times T}$$

Here $\mathbf{A}$ is the *workload matrix* representing $\mathcal{A}$. At iteration $t$, $\mathcal{A}$ can only use the current and previous gradients, so $a_{tj}=0$ for $j>t$ (ie. $\mathbf{A}$ is lower-triangular). In this notation, the iterates of $\mathcal{A}$ satisfy

$$\mathbf{X} = \mathbf{X}_0 - \gamma\mathbf{A}\mathbf{G}. \tag{2}$$

**Example 2.1** (SGD)**.** Define the *prefix-sum* matrix $\mathbf{S}\in\mathbb{R}^{T\times T}$ as the all-ones lower-triangular matrix. If $\mathbf{A}=\mathbf{S}$, then (2) is simply SGD with learning rate $\gamma$. As discussed by Denisov et al. [10, Section 4], we also recover SGD with momentum using an appropriate transformation $\mathbf{S}'$ of $\mathbf{S}$.

### 2.1 Matrix Factorization and Privacy Mechanisms

In order to make the output of (2) differentially private, we typically need to clip the gradients and add noise. Let $\overline{\mathbf{G}}$ denote the matrix whose rows (gradients) have been clipped to some $\ell_2$ threshold $\alpha$. Let $\mathbf{Z}\in\mathbb{R}^{T\times d}$ be a matrix with entries drawn independently from $\mathcal{N}(0,\varsigma^2/d)$. The well-known DP-SGD algorithm [1] adds this noise to each clipped gradient, so that

$$\mathbf{X} = \mathbf{X}_0 - \gamma\mathbf{A}(\overline{\mathbf{G}} + \mathbf{Z}). \tag{3}$$

For consistency, we consider (2) to be the special case of (3) where $\mathbf{Z}=\mathbf{0}$ and $\alpha=\infty$. The variance $\varsigma^2$ depends on the clipping threshold $\alpha$ and desired $(\varepsilon,\delta)$ privacy we aim to achieve [1].

To derive algorithms with improved DP guarantees, Denisov et al. [10] add the noise $\mathbf{Z}$ to a factorized version of $\mathbf{A}$. For a factorization $\mathbf{A} = \mathbf{BC}$ with $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{T\times T}$, we add noise to the iterates via:

$$\mathbf{X} = \mathbf{X}_0 - \gamma\mathbf{B}\left(\mathbf{C}\overline{\mathbf{G}} + \mathrm{sens}(\mathbf{C})\mathbf{Z}\right) \equiv \mathbf{X}_0 - \gamma\left(\mathbf{A}\overline{\mathbf{G}} + \mathrm{sens}(\mathbf{C})\mathbf{B}\mathbf{Z}\right). \tag{4}$$

Here, $\mathrm{sens}(\mathbf{C})$ is a number representing the sensitivity of the mapping $\overline{\mathbf{G}} \mapsto \mathbf{C}\overline{\mathbf{G}}$ to "adjacent" input changes. We note that the sensitivity changes depending on the notion of adjacency. In single-epoch settings, two input matrices are adjacent if they differ by a single row [10], so the sensitivity function is $\mathrm{sens}(\mathbf{C}) := \max_{i\in\{1,\dots,T\}} \|\mathbf{C}_{[:,i]}\|_2$, i.e. the maximum $\ell_2$-squared column norm of $\mathbf{C}$. For details and extensions to multiple epochs, see [7].

If the variance of entries of $\mathbf{Z}$ is fixed to some value $\zeta^2/d$, then for all the possible factorizations $\mathbf{A} = \mathbf{BC}$ in (4) have exactly same privacy guarantees, depending only on $\zeta$. It will also be convenient to define $\sigma = \mathrm{sens}(\mathbf{C})\zeta$ as the 'effective' variance of $\mathbf{Z}$ after re-scaling by the sensitivity. Note that for a fixed $\sigma$, the privacy guarantees of (4) might be different depending on the sensitivity.

The factorization $\mathbf{B} = \mathbf{A}, \mathbf{C} = \mathbf{I}$ recovers DP-SGD (3), but factorizations with better privacy-utility trade-offs may exist. The formulation of Eq. (4) transfers the linear optimization algorithm (2) into the setting of the matrix mechanism [26], a well-studied family of mechanisms in differential privacy. Denisov et al. [10], Choquette-Choo et al. [7] show that the mechanism in Eq. (4) provides a DP guarantee equivalent to a single application of the Gaussian mechanism, which can be computed tightly using numerical accounting techniques [49, 24].

**Finding good factorizations.** Intuitively, a factorization $\mathbf{A} = \mathbf{BC}$ is good if $\mathrm{sens}(\mathbf{C})$ is small and the added noise $\mathbf{BZ}$ does not significantly degrade the convergence of (4). In order to quantify the effect of this added correlated noise on optimization, Denisov et al. [10] derive an online regret bound for (4) in the convex case against an adaptive adversary. Translating this via online-to-batch convergence to the stochastic setting, the iterates $\mathbf{x}_t$ satisfy

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right] \leq \mathcal{O}\left(\frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\gamma T} + \gamma\tilde{L}^2 + \gamma\zeta\tilde{L}\frac{\mathrm{sens}(\mathbf{C})\,\|\mathbf{B}\|_F}{\sqrt{T}}\right) \tag{5}$$

where $\tilde{L}$ is the Lipshitz constant of $f$. Denisov et al. [10] therefore use $\mathrm{sens}(\mathbf{C})\,\|\mathbf{B}\|_F$ as a proxy for the impact of the factorized noise scheme on convergence. To find factorizations with good convergence properties, Denisov et al. [10], Choquette-Choo et al. [7] minimize $\mathrm{sens}(\mathbf{C})\,\|\mathbf{B}\|_F$ subject to the constraint $\mathbf{A} = \mathbf{BC}$, which is equivalent to the following objective:

*Problem* 2.2 (Minimal-Norm Matrix Factorization). Given a lower triangular matrix $\mathbf{A} \in \mathbb{R}^{T\times T}$, define $\mathrm{OPT}_F(\mathbf{A}) = (\mathbf{B}, \mathbf{C})$, where $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{T\times T}$ solve the following optimization problem.

$$\min_{\mathbf{B},\mathbf{C}} \|\mathbf{B}\|_F^2 \quad \text{such that } \mathbf{BC} = \mathbf{A},\ \mathrm{sens}(\mathbf{C}) = 1. \tag{6}$$

Eq. (6) is well-studied in the privacy literature and can be solved with a variety of numerical optimization algorithms [51, 30, 10, 7]. We also note that Denisov et al. [10] show that without loss of generality, we can assume $\mathbf{B}$ and $\mathbf{C}$ are lower triangular.

**Finding improved factorizations.** We argue that (5) is pessimistic in stochastic settings. For SGD (when $\mathbf{B} = \mathbf{A}$), the last term in (5) is $\mathcal{O}(\gamma\,\mathrm{sens}(\mathbf{C})\zeta\tilde{L}\sqrt{T})$, which diverges with $T$ for a constant stepsize. However, under the same assumptions as in [10], SGD with constant stepsize actually achieves a faster rate of $\mathcal{O}(\gamma\,\mathrm{sens}(\mathbf{C})\zeta\tilde{L})$ (see [41]).

In this paper, we turn our attention to the *smooth functions* in order to focus on non-convex functions. We show in Appendix A, there are matrices $\mathbf{B}_1, \mathbf{B}_2$ such that $\mathrm{sens}(\mathbf{C}_1)\|\mathbf{B}_1\|_F = \mathrm{sens}(\mathbf{C}_2)\|\mathbf{B}_2\|_F$, but Eq. (4) diverges with $\mathbf{B}_1$ and converges with $\mathbf{B}_2$, therefore showing that Frobenius norm is not the right measure in the smooth case as well.

This begs the question of whether there are objectives that better capture the impact of the noise injected in (4) on convergence. To answer this, we derive a bound that can exhibit better dependence on $\mathbf{B}$ to design better factorizations for differentially private optimization.

## 3 Problem Formulation

To study the effect of the noise $\mathbf{BZ}$ on optimization, we analyze a slightly simplified objective that omits parts of (4) not directly related to linear noise correlation. We do this as follows:

(I) We assume that each $\mathbf{g}_t$ is the true gradient at the point $\mathbf{x}_t$, i.e. $\mathbf{g}_t = \nabla f(\mathbf{x}_t)$.

(II) We omit gradient clipping from our analysis. Alternatively, we can view this as setting the clipping threshold $\alpha = \infty$ so that $\overline{\mathbf{G}} = \mathbf{G}$ in (4).

(III) We restrict the class $\mathcal{A}$ to SGD-type algorithms where $\mathbf{A} = \mathbf{S}$, as in Example 2.1.

We impose (I) for simplicity of presentation. Our results can be extended to stochastic gradients in a direct fashion. Restriction (II) is also for simplicity. First, clipping is not directly applied to the noise $\mathbf{BZ}$. Second, for bounded domains or Lipschitz $f$, our analysis still holds with clipping. Last, practical DP methods often use adaptive clipping [45] instead of fixed clipping. We are not aware of convergence analyses for such schemes. We impose (III) in order to limit the class of algorithms $\mathcal{A}$ to a well-understood subclass. The convergence properties of (2) for general matrices $\mathbf{A}$ are not well-understood even when there is no noise ($\mathbf{Z} = \mathbf{0}$). As we discuss in Section 4, even with these simplifications, the effect of $\mathbf{BZ}$ is not well-understood.

Due to (III), we study factorizations $\mathbf{BC}$ of the matrix $\mathbf{A} = \mathbf{S}$, as in Example 2.1. Then, (4) becomes

$$\mathbf{X} = \mathbf{X}_0 - \gamma \left(\mathbf{SG} + \mathrm{sens}(\mathbf{C})\mathbf{BZ}\right). \tag{7}$$

In vector notation, for $\mathbf{b}_0 = \mathbf{0}$ and $\mathbf{B} = [\mathbf{b}_1, \dots \mathbf{b}_T]^\top$,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \left[\nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_t)^\top \mathbf{Z}\right], \tag{8}$$

where for simplicity of presentation, we re-scaled the noise $\mathbf{Z}$ by the sensitivity, $\sigma^2 = \mathrm{sens}^2(\mathbf{C})\zeta^2$. We now discuss several noteworthy special cases of (8).

**Example 3.1** (PGD)**.** If $\mathbf{B} = \mathbf{S}$ (see Example 2.1) we recover SGD with uncorrelated additive noise, also known as *perturbed gradient descent* (PGD), where

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \left[\nabla f(\mathbf{x}_t) + \mathbf{z}_{t+1}\right]. \tag{9}$$

The convergence rate of SGD (and therefore PGD) is well-understood in the optimization literature (e.g. see Bubeck [6, Section 6]).

**Example 3.2** (Anti-PGD)**.** By setting $\mathbf{B} = \mathbf{I}$, we get an algorithm that at every iteration adds an independent noise vector $\mathbf{z}_{t+1}$ and subtracts the previously added noise $\mathbf{z}_t$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \left[\nabla f(\mathbf{x}_t) + \mathbf{z}_{t+1} - \mathbf{z}_t\right], \;\; \mathbf{z}_0 = \mathbf{0} \tag{10}$$

Intuitively, this removes some of the noise added in the prior round. This is (up to a learning rate factor) the *anti-correlated perturbed gradient descent* (Anti-PGD) method proposed by Orvieto et al. [37], who study its generalization properties. Anti-PGD is also equivalent to SGD with randomized-smoothing [11]. The equivalence follows from defining $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \gamma \mathbf{z}_t$ and rewriting (10) as

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \nabla f(\tilde{\mathbf{x}}_t - \gamma \mathbf{z}_t).$$

While randomized smoothing algorithm is popular for non-smooth optimization, Vardhan and Stich [47] analyze its convergence properties in the smooth non-convex setting.

**Example 3.3** (Tree Aggregation DP-FTRL)**.** For $k \geq 1$ and $t = 2^{k-1}$, define $\mathbf{H}_k \in \mathbb{R}^{(2^k-1)\times t}$ recursively as follows:

$$\mathbf{H}_1 = (1)\,, \; \mathbf{H}_{k+1} = \begin{pmatrix} \mathbf{H}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_k \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$$

where $\mathbf{1}$ above represents an all-ones row of appropriate width. For $T = 2^{k-1}$, if $\mathbf{C} = \mathbf{H}_k$ and $\mathbf{B} = \mathbf{S}\mathbf{C}_k^\dagger$ where $\mathbf{C}_k^\dagger$ denotes a carefully chosen right pseudo-inverse of $\mathbf{C}$, then we recover the same noise matrix $\mathbf{B}$ as in the DP-FTRL algorithm with either the online or full Honaker estimator (depending on the choice of $\mathbf{C}^\dagger$) as in [20, 10]. Note that $\mathbf{B}, \mathbf{C}$ are not square. This can be remedied by appropriately projecting onto $\mathbb{R}^T$. See Choquette-Choo et al. [7, Appendix D.3] for details.

## 4 Deriving Tighter Convergence Rates

We would like convergence rates for (7) that apply to any factorization and yield tight convergence rates for notable special cases. We pay special attention to PGD (Example 3.1) and Anti-PGD (Example 3.2), as they represent extremes in the space of factorizations ($\mathbf{S} = \mathbf{SI}$ and $\mathbf{S} = \mathbf{IS}$, respectively). As we will show, it is possible to use existing theoretical tools to derive tight convergence rates for both, *but not simultaneously*.

Below, we discuss ways to derive tight rates for PGD and Anti-PGD, and how these rates involve incompatible analyses. We then develop a novel analytic framework involving *restart iterates* that allows us to analyze both methods simultaneously, as well as (7) for general factorizations. We start by formally stating our assumptions. For simplicity of presentation, we re-scale the noise $\mathbf{Z}$ by the sensitivity of $\mathbf{C}$, i.e. $\sigma^2 = \mathrm{sens}^2(\mathbf{C})\zeta^2$; we will suppress the $\mathbf{C}$ dependence of $\sigma$.

**Assumption 4.1** (Noise)**.** The rows $\mathbf{z}_1, \ldots, \mathbf{z}_T$ of the noise matrix $\mathbf{Z}$ are independent random vectors such that $\forall t$, $\mathbb{E}[\mathbf{z}_t] = \mathbf{0}$ and $\mathbb{E}\left\|\mathbf{z}_t\right\|^2 \leq \sigma^2$.

We do not assume $\tilde{L}$-Lipshitzness in our results, but we do assume $L$-smoothness. This is a relatively standard assumption in optimization literature [6].

**Assumption 4.2** ($L$-smoothness)**.** The function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable, and there exists $L > 0$ such that for all $x, y \in \mathbb{R}^d$, $\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\| \leq L\left\|\mathbf{x} - \mathbf{y}\right\|$.

For *some* of the results we will assume convexity.

**Assumption 4.3** (Convexity)**.** The function $f : \mathbb{R}^d \to \mathbb{R}$ is convex, i.e. $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $f(\mathbf{x}) - f(\mathbf{y}) \leq \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle$. When assuming convexity, we also assume the infimum of $f$ is achieved in $\mathbb{R}^d$.

### 4.1 Convergence Rates for PGD and Anti-PGD

In this section we discuss the (distinct) convergence analyses of PGD and Anti-PGD, and the suboptimal results derived by trying to apply the proof technique for one to the other. We focus on the convex setting for brevity, though these analyses can be directly extended to the non-convex setting.

**PGD.** The convergence of PGD (Example 3.1) is well-understood since it is a special case of SGD. One can show the following.

**Proposition 4.4** (Adapted from Dekel et al. [9, Theorem 1])**.** *Under Assumptions 4.1, 4.2 and 4.3, if* $\mathbf{B} = \mathbf{S}$ *and* $\gamma < 1/2L$*, then the output of* (7) *satisfies*

$$\sum_{t=0}^{T} \frac{\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right]}{T+1} \leq \mathcal{O}\left(\frac{\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|^2}{\gamma T} + \gamma\sigma^2\right). \tag{11}$$

The proof follows from combining the update (9), standard facts about convex functions, and the fact that $\gamma < 1/2L$, to get the inequality

$$\mathbb{E}_t\left\|\mathbf{x}_{t+1} - \mathbf{x}^\star\right\|^2 \leq \left\|\mathbf{x}_t - \mathbf{x}^\star\right\|^2 - \gamma\left(f(\mathbf{x}_t) - f^\star\right) + \gamma^2\sigma^2.$$

It is left to average over iterations $0 \leq t \leq T$.

**Anti-PGD.** For Anti-PGD (Example 3.2), one can show the following.

**Proposition 4.5.** *Under Assumptions 4.1, 4.2 and 4.3, if* $\mathbf{B} = \mathbf{I}$ *and* $\gamma < 1/2L$*, then the output of* (7) *satisfies*

$$\sum_{t=0}^{T} \frac{\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right]}{T+1} \leq \mathcal{O}\left(\frac{\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|^2}{\gamma T} + L\gamma^2\sigma^2\right) \tag{12}$$

Since $L\gamma < 1/2$, the RHS of (12) is strictly smaller than the RHS of (11). While this result may be known, we were unable to find a reference, so we provide a complete proof in Appendix D. The proof utilizes perturbed iterate analysis [28]. We define a *virtual sequence* $\{\tilde{\mathbf{x}}_t\}_{t=0}^T$ as follows:

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma\nabla f(\mathbf{x}_t), \qquad \tilde{\mathbf{x}}_0 = \mathbf{x}_0 \tag{13}$$

The $\tilde{\mathbf{x}}_t$ are the iterates of (7) when $\mathbf{Z} = \mathbf{0}$. We can then prove the following descent inequality:

$$\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 \leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - \frac{\gamma}{2}\left(f(\mathbf{x}_t) - f^\star\right) + 2L\gamma\,\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|^2.$$

Because of the anti-correlation in (10), the virtual iterates $\tilde{\mathbf{x}}_t$ are close to the real iterates $\mathbf{x}_t$, as $\mathbf{x}_t - \tilde{\mathbf{x}}_t = \gamma \mathbf{z}_t$. Averaging over $t$, we recover (12). See Appendix D for details.

**Tightness.** The noise terms (those terms involving $\sigma^2$) in (11), (12) are both tight. We show this in Appendix E on the objective $f(\mathbf{x}) = (L/2)\,\|\mathbf{x}\|^2$.

**Difficulties in a unified analysis.** The proof techniques for PGD and Anti-PGD above are notably different, and as we explain in Appendix F, do not lead to favorable results when trying to use one of the two strategies to analyze both.

### 4.2 Main Results and Analytic Techniques

To unify the proof techniques above, we use a modified virtual sequence with *restart iterations*. For a parameter $\tau = \tilde{\Theta}(1/L\gamma)$ (throughout, $\tilde{\mathcal{O}}$ and $\tilde{\Theta}$ hide poly-logarithmic factors), we define

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) && \text{if } t+1 \neq 0 \bmod \tau \\ \tilde{\mathbf{x}}_{t+1} &= \mathbf{x}_{t+1} && \text{if } t+1 = 0 \bmod \tau. \end{aligned} \tag{14}$$

Similar to the virtual sequence in (13), $\tilde{\mathbf{x}}_t$ incorporates only deterministic gradients $\nabla f(\mathbf{x}_t)$. However, every $\tau$ iterations we reset $\tilde{\mathbf{x}}_t$ to the real iterate $\mathbf{x}_t$. This allows us to control the divergence between the virtual sequence and the real sequence (enabling a tight analysis of PGD), while still capturing the convergence benefits of anti-correlated noise (enabling a tight analysis of Anti-PGD).

The parameter $\tau$ is independent of $\mathbf{B}$, and depends only on the geometry of $f$ and the stepsize $\gamma$. Using this machinery, we can prove convergence rates of (7) for *any* factorization $\mathbf{S} = \mathbf{BC}$. These rates involve $\ell_2$ distances between the rows $\mathbf{b}_t$ of the matrix $\mathbf{B}$ (where $\mathbf{b}_0 = \mathbf{0}$ for convenience).

**Theorem 4.6** (non-convex)**.** *Suppose Assumptions 4.1 and 4.2 hold, $\gamma \leq 1/4L$, and $\tau = 1/\gamma L$. Then* (7) *produces iterates whose average error $(T+1)^{-1}\sum_{t=0}^{T}\mathbb{E}\,\|\nabla f(\mathbf{x}_t)\|^2$ is upper bounded by*

$$\mathcal{O}\left(\frac{(f(\mathbf{x}_0) - f^\star)}{\gamma T} + \frac{\sigma^2}{T\tau} \times \left[\frac{1}{\tau}\sum_{t=1}^{T}\left\|\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau}\right\|^2 + \sum_{\substack{1 \leq t \leq T \\ t=0 \bmod \tau}} \|\mathbf{b}_t - \mathbf{b}_{t-\tau}\|^2\right]\right).$$

**Theorem 4.7** (convex)**.** *Under Assumptions 4.1, 4.2, and 4.3, if $\gamma \leq 1/4L$ and $\tau = \tilde{\Theta}(1/\gamma L)$, then* (7) *produces iterates with average error $(T+1)^{-1}\sum_{t=0}^{T}\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right]$ upper bounded by*

$$\tilde{\mathcal{O}}\left(\frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\gamma T} + \frac{\sigma^2}{TL\tau} \times \left[\frac{1}{\tau}\sum_{t=1}^{T}\left\|\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau}\right\|^2 + \sum_{\substack{1 \leq t \leq T \\ t=0 \bmod \tau}} \|\mathbf{b}_t - \mathbf{b}_{t-\tau}\|^2 + \left\|\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau}\right\|^2\right]\right).$$

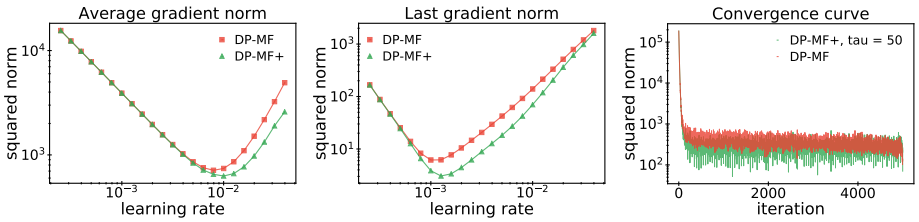
We give complete proofs in Appendix C. These convergence rates consist of two terms: The first term states how fast the function would converge in the absence of the noise. The second term, the *noise term*, is the focus of our paper, as it shows how the correlated noise $\mathbf{BZ}$ affects convergence.

These rates involve only differences of rows of $\mathbf{B}$ that are at most $\tau$ iterations apart. Intuitively, $\tau$ is a coarse indicator of whether an iterate $\mathbf{x}_t$ is still sensitive to the noise injected at an iteration $t' < t$. If $t > t' + \tau$, then changes in the noise added at step $t$ are effectively uncorrelated to iteration $t'$. As we detail in Appendix, applying Theorem 4.7 to the special cases in Examples 3.1, 3.2 recovers their tight convergence rates in (11), (12) correspondingly.

## 5 Finding Better Factorizations

We now draw on our results in Section 4 to develop better mechanisms for the MF-DP-FTRL framework. We modify the objective underlying the offline matrix factorization problem during the first stage of the MF-DP-FTRL workflow (Fig. 1). Specifically, observe that the noise term in Theorems 4.6 and 4.7 can be rewritten in matrix notation (up to multiplicative constants) as

$$\|\mathbf{\Lambda}_\tau \mathbf{B}\|_F^2 = \sum_{t=1}^{T}\left\|\boldsymbol{\lambda}_t^\top \mathbf{B}\right\|^2 = \sum_{\substack{1 \leq t \leq T \\ t=0 \bmod \tau}} \|\mathbf{b}_t - \mathbf{b}_{t-\tau}\|^2 + \sum_{\substack{1 \leq t \leq T \\ t \neq 0 \bmod \tau}} \left\|\frac{1}{\sqrt{\tau}}\left(\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau}\right)\right\|^2 \tag{15}$$




(a) Average gradient norm for varying learning rates.

(b) Last gradient norm for varying learning rates.

(c) Gradient norm over time for $\gamma = 10^{-2}$.

Figure 2: Comparison of the average and last gradient norms for DP-MF and DP-MF$^+$ on a random non-strongly convex quadratic function with $L = 10$.

where $\mathbf{\Lambda}_\tau = \left[\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_T^\top\right]^\top \in \mathbb{R}^{T \times T}$, and we set the rows $\boldsymbol{\lambda}_t$ appropriately to select corresponding row differences of $\mathbf{B}$ with either coefficient 1 or $1/\sqrt{\tau}$ depending on the index $t$. We give a precise definition of $\mathbf{\Lambda}_\tau$ and an explicit example when $T = 12, \tau = 3$ in Appendix B.

Recall that [10] minimize the Frobenius norm objective (6) based on their derived convergence bounds in (5). Since our derived convergence bounds are strictly tighter, we propose using Eq. (15) as the new objective function in (6). Intuitively, since $\|\mathbf{\Lambda}_\tau \mathbf{B}\|_F^2$ is a better proxy for learning performance than $\|\mathbf{B}\|_F^2$, minimizing this quantity in the offline factorization problem should lead to ERM methods with better privacy-utility trade-offs.

We can solve our new offline matrix factorization problem in a straightforward manner. We can show that for $\mathbf{A} = \mathbf{S}$, we can solve this modified problem by first computing the solution $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ using $\mathrm{OPT}_F(\mathbf{\Lambda}_\tau \mathbf{A})$. The solution to our modified objective is then $\mathbf{C} = \tilde{\mathbf{C}}$, $\mathbf{B} = \mathbf{A}\mathbf{C}^{-1}$. This implies we can use existing open-source solvers designed for (6) [51, 30, 10].

## 6 Experiments

In this section, we evaluate the ERM performance of MF-DP-FTRL under different offline factorization objectives. We focus on the Frobenius norm objective (6), which we refer to as DP-MF [10, 7], and our modified objective (15), which we refer to as DP-MF$^+$.

### 6.1 Validating Theoretical Results

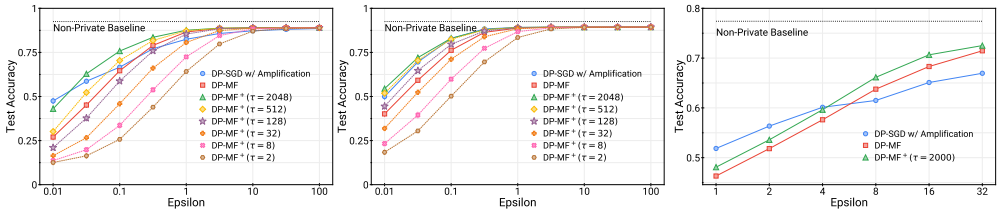
We first validate our theoretical results above by comparing the convergence of DP-MF and DP-MF$^+$ on a *random quadratic* function that satisfies the assumptions of Theorem 4.7. Notably, we ensure the quadratic is not strongly convex. We treat $\tau$ in (15) as a hyperparameter and tune it over a fixed grid. For complete details, please refer to Appendix H. We present the results in Fig. 2.

In Fig. 2(a) we plot $\frac{1}{T}\sum_{t=0}^{T} \|\nabla f(\mathbf{x}_t)\|^2$, as this quantity is proportional to the LHS of Theorem 4.7. For all learning rates, DP-MF$^+$ either matches or outperforms DP-MF. Moreover, the advantage of DP-MF$^+$ increases as the learning rate increases. This corresponds to our theory in Theorem 4.7. Indeed, the larger the stepsize $\gamma$, the smaller the optimal $\tau$ (as $\tau = \Theta\left(1/\gamma L\right)$), and the more often restarts are used in the analysis of Theorem 4.7.

Fig. 2(b) further depicts the last-iterate behaviours of DP-MF and DP-MF$^+$, which is often more practically relevant. Interestingly, the last iterate behaviour is improved even in the cases where the average behaviour does not improve. Finally, in Fig. 2(c) we pick $\gamma = 10^{-2}, \tau = 50$ as the parameters for which both the average and the last-iterate behaviours are improved and plot the convergence curve over iterations. DP-MF$^+$ has regular oscillating behaviour, allowing it to achieve a good final-iterate performance. The period of these oscillations is exactly equal to $\tau$.

### 6.2 Practical DP Training Experiments

We now compare DP-MF, DP-MF$^+$, and DP-SGD with privacy amplification [1] on the MNIST, CIFAR-10, and Stack Overflow datasets. We omit from comparison DP-FTRL [20] and DP-Fourier




(a) MNIST, 1 epoch (b) MNIST, 16 epochs (c) CIFAR-10, 20 epochs

Figure 3: Test set accuracy of various mechanisms on the MNIST and CIFAR-10 datasets.

[7] as these methods are strictly dominated by DP-MF. Unlike our theoretical analysis, we include clipping to derive formal $(\varepsilon, \delta)$ privacy guarantees. To facilitate a fair comparison, we set $\delta = 10^{-6}$ in all the settings, and compare against varying $\varepsilon$. We give complete experimental details in Appendix H

**MNIST, logistic regression.** We train for $T = 2048$ iterations and either 1 or 16 epochs depending on the batch size, corresponding to a batch size of 29 and 469 respectively.[2] We fix the clipping threshold at 1.0 and the learning rate at 0.5. We vary $\tau$ in (15) over $\{2, 8, 32, 128, 512, 2048\}$. The results are in Figs. 3(a) and 3(b). DP-MF$^+$ improves monotonically with $\tau$, performing best when $\tau = 2048 = T$. For such $\tau$, DP-MF$^+$ consistently out-performs DP-MF across all settings. Recall from (15) that this corresponds to the offline objective $\|\mathbf{\Lambda}_T\mathbf{B}\|_F^2$ where $\lambda_{ii} = 1/\sqrt{T}$ for all $i < T$ and $\lambda_{TT} = 1$. This objective strongly penalizes errors on the final iterate, which is the model used to compute test accuracy.

We also see that DP-MF$^+$ expands the number of settings in which we can beat DP-SGD. DP-MF only outperforms DP-SGD for sufficiently large $\varepsilon$ ($\varepsilon \geq 0.31$ for 1 epoch and $\varepsilon \geq 31$ for 16 epochs). By contrast, DP-MF$^+$ outperforms DP-SGD in every setting except when $\varepsilon = 0.01$ and 1 epoch. None of the mechanisms reached the accuracy levels obtained by the non-private baseline, even at $\varepsilon = 100$. We suspect this is due to the fact that we are using a fixed but aggressive clipping threshold of 1.0 across all experiments, which helps in the moderate privacy regime but hurts in very low privacy regime. Even though DP-MF$^+$ does not use privacy amplification, it outperforms DP-SGD, which uses privacy amplification. This is due to the efficient noise anti-correlations in DP-MF$^+$. If amplification were not possible, performance of DP-SGD would degrade even further.

**CIFAR-10, CNN.** We follow the experimental setup from [7]. Specifically, we train all mechanisms for 20 epochs and $T = 2000$ iterations, which corresponds to a batch size of 500.[3] We tune the learning rate over a fixed grid. We fix $\tau = T = 2000$ in DP-MF$^+$ as we found that worked best in the MNIST experiments. The results are given in Fig. 3(c). We see that DP-MF$^+(\tau = 2000)$ offers a consistent improvement over DP-MF across all choices of $\varepsilon$ considered. Both DP-MF and DP-MF$^+$ beat DP-SGD for $\varepsilon > 4$. This observation is consistent with prior work on DP-FTRL and DP-MF, where DP-SGD performs relatively better with smaller $\varepsilon$ while DP-MF performs better with larger $\varepsilon$.

**Stack Overflow, LSTM.** In Appendix H, we compare DP-MF and DP-MF$^+$ on a federated learning task with *user-level* differential privacy. We do not compare DP-SGD on this task, as amplification techniques such as shuffling and subsampling are not possible in practical federated learning settings [20]. In this task, we train an LSTM network to do next-word prediction on the Stack Overflow dataset. To be consistent with the prior work [10] and to test if our proposed factorizations are compatible with the other types of workloads $\mathbf{A}$ from Eq. (2), we use momentum and learning rate decay. Our results are given in Table 2. We see that two methods perform comparably, verifying competitiveness of our method. Note that this task uses federated averaging [32] instead of gradient

[2] In practice, one often trains small-scale models for many epochs, perhaps even using full-batch gradients, to improve the privacy/utility trade-off (at the cost of increased computation). We are interested in the *relative* performance for a fixed computation budget, so we train for a small number of epochs.

[3] While Choquette-Choo et al. [7] use momentum and learning rate decay, we omit the use of such techniques as they are orthogonal to our theoretical results.

descent. Developing offline factorization objectives specifically for federated learning remains an open problem.

## 7 Conclusion

In this work, we developed analytic techniques to study the convergence of gradient descent under linearly correlated noise that is motivated from a class of DP mechanisms. We derived tighter bounds than currently exist in the literature, and we use our novel theoretical understanding to design privacy mechanisms with improved convergence. Perhaps more importantly, our work highlights the wealth of stochastic optimization questions arising from recent advances in differentially private model training. As such, we distill and formalize various optimization problems arising from recent work on matrix mechanisms for DP. Our work raises a host of questions and open problems, including extending our analysis to include things such as clipping, shuffling, and momentum. Another key extension is to derive last-iterate convergence rates rather than average-iterate convergence rates, as in some settings it is only the final "released" model that needs formal privacy guarantees. Given the improved generalization properties of Anti-PGD [37], one could also investigate how to design more general linearly correlated noise mechanisms which improve both privacy and generalization.

## 8 Acknowledgments

The authors would like to thank Francesco D'Angelo, Nina Mainusch and Linara Adylova for their comments on the manuscript. The authors would also like to thank the reviewers for their helpful suggestions in improving the clarity of the writing.

## References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2016. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

[2] Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/f0e52b27a7a5d6a1a87373dffa53dbe5-Paper.pdf.

[3] The TensorFlow Federated Authors. TensorFlow Federated Stack Overflow dataset, 2019. URL https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data.

[4] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[5] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

[6] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, nov 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL https://doi.org/10.1561/2200000050.

[7] Christopher A. Choquette-Choo, H. Brendan McMahan, Keith Rush, and Abhradeep Thakurta. Multi-epoch matrix factorization mechanisms for private machine learning, 2022. URL https://arxiv.org/abs/2211.06530.

[8] Rudrajit Das, Satyen Kale, Zheng Xu, Tong Zhang, and Sujay Sanghavi. Beyond uniform Lipschitz condition in differentially private optimization. *arXiv preprint arXiv:2206.10713*, 2022.

[9] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *J. Mach. Learn. Res.*, 13(null):165–202, jan 2012. ISSN 1532-4435.

[10] Sergey Denisov, Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In *Neural Information Processing Systems*, 2022.

[11] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[12] Sanghamitra Dutta, Gauri Joshi, Soumyadip Ghosh, Parijat Dube, and Priya Nagpurkar. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed sgd. In *International conference on artificial intelligence and statistics*, pages 803–812. PMLR, 2018.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

[14] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. The power of factorization mechanisms in local and central differential privacy. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 425–438, 2020.

[15] 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.

[16] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 954–964. IEEE, 2022.

[17] Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly converging error compensated sgd. *Advances in Neural Information Processing Systems*, 33: 20889–20900, 2020.

[18] Monika Henzinger and Jalaj Upadhyay. Constant matters: Fine-grained complexity of differentially private continual observation using completely bounded norms. Cryptology ePrint Archive, Paper 2022/225, 2022. URL `https://eprint.iacr.org/2022/225`. `https://eprint.iacr.org/2022/225`.

[19] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

[20] Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/kairouz21b.html`.

[21] 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.

[22] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[23] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[24] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using FFT. In *International Conference on Artificial Intelligence and Statistics*, pages 3358–3366. PMLR, 2021.

[25] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 123–134, 2010.

[26] Chao Li, Gerome Miklau, Michael Hay, Andrew Mcgregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB Journal*, 24:757–781, 2015.

[27] Aurelien Lucchi, Frank Proske, Antonio Orvieto, Francis Bach, and Hans Kersting. On the theoretical properties of noise correlation in stochastic optimization. *Neural Information Processing Systems*, 2022.

[28] Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017. doi: 10.1137/16M1057000. URL `https://doi.org/10.1137/16M1057000`.

[29] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *arXiv preprint arXiv:1808.03537*, 2018.

[30] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Hdmm: Optimizing error of high-dimensional statistical queries under differential privacy. *arXiv preprint arXiv:2106.12118*, 2021.

[31] Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees. *Google AI Blog*, 2022.

[32] 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.

[33] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33: 17309–17320, 2020.

[34] Aritra Mitra, Rayana Jaafar, George J Pappas, and Hamed Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34:14606–14619, 2021.

[35] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[36] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 3581–3607. PMLR, 2022.

[37] Antonio Orvieto, Hans Kersting, Frank Proske, Francis Bach, and Aurelien Lucchi. Anticorrelated noise injection for improved generalization. *arXiv preprint arXiv:2202.02831*, 2022.

[38] Antonio Orvieto, Anant Raj, Hans Kersting, and Francis Bach. Explicit regularization in overparametrized models via noise injection. *arXiv preprint arXiv:2206.04613*, 2022.

[39] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(64)90137-5. URL `https://www.sciencedirect.com/science/article/pii/0041555364901375`.

[40] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

[41] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Annual Conference Computational Learning Theory*, 2009.

[42] Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1g2JnRcFX.

[43] Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed updates. *J. Mach. Learn. Res.*, 21(1), jun 2022. ISSN 1532-4435.

[44] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. *Advances in Neural Information Processing Systems*, 31, 2018.

[45] Om Thakkar, Galen Andrew, and H. B. McMahan. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, 2021.

[46] Hoang Tran and Ashok Cutkosky. Momentum aggregation for private non-convex erm, 2022.

[47] Harsh Vardhan and Sebastian U. Stich. Tackling benign nonconvexity with smoothing and stochastic gradients, 2022. URL https://arxiv.org/abs/2202.09052.

[48] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[49] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[50] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019.

[51] Ganzhao Yuan, Yin Yang, Zhenjie Zhang, and Zhifeng Hao. Convex optimization for linear query processing under approximate differential privacy, 2016. URL https://arxiv.org/abs/1602.04302.

[52] Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. *Advances in Neural Information Processing Systems*, 33:5332–5344, 2020.

[53] Chulhee Yun, Shashank Rajput, and Suvrit Sra. Minibatch vs local SGD with shuffling: Tight convergence bounds and beyond. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LdlwbBP2mlq.

[54] Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Toward understanding the importance of noise in training neural networks. In *International Conference on Machine Learning*, pages 7594–7602. PMLR, 2019.

[55] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled Rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642. PMLR, 2019.

# A Additional Examples

## A.1 Why the Frobenius Norm is not Predictive

In this section we give an explicit example of a matrix $\mathbf{B}$ for which the Frobenius norm $\|\mathbf{B}\|_F$ does not give a good estimation of the optimization behavior of (7).

**Example A.1** (Chess-PGD)**.** We consider the special case of algorithm (7) whose noise correlation matrix $\mathbf{B}$ whose lower triangle has a chess board-like structure given by

$$\mathbf{B}_{\text{chess}} = \sqrt{2}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & & & & \\ 0 & 1 & 0 & \dots & 1 \end{pmatrix}$$

We refer to this algorithm (whose perturbed noise structure is given by $\mathbf{B}_{\text{chess}}$) as Chess-PGD. Note that $\text{sens}(\mathbf{C}_{\text{chess}})\,\|\mathbf{B}_{\text{chess}}\|_F = \text{sens}(\mathbf{C}_{\mathbf{S}})\,\|\mathbf{S}\|_F$. Despite this, PGD (for which $\mathbf{B} = \mathbf{S}$) converges strictly faster than Chess-PGD in Fig. 4.

By contrast, our Theorem 4.7 is better able to capture the behaviour of Chess-PGD. Suppose that $\tau \leq T/4$. Given a row $\mathbf{b}_t$ of $\mathbf{B}_{\text{chess}}$, for any $t' < t$ we have

$$\frac{t - t'}{2} \leq \|\mathbf{b}_t - \mathbf{b}_{t'}\|^2 \leq t.$$

Therefore, at least $T/4$ of the summands in the noise term of Theorem 4.7 are on the order of $\Theta(T)$. Plugging in this estimate into the convergence rate, we find that Chess-PGD produces iterates that satisfy the convergence rate

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[f(\mathbf{x}_t) - f^*\right] = \tilde{\mathcal{O}}\left(\frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\gamma T} + LT\gamma^2\sigma^2\right). \tag{16}$$

Indeed, as we show below (and plot in Figure 4), Chess-PGD linearly diverges with $T$ as predicted.

## A.2 Experimental Comparison of PGD with Chess-PGD

In this section we illustrate that Chess-PGD diverges while PGD converges for the same quadratic functions as in Section 6. We set the stepsize constant, $\gamma = 0.02$. We plot $\|\nabla f(\mathbf{x}_t)\|^2$ at each iteration $t$. We see that, as predicted by (16), Chess-PGD diverges with linear rate in $T$, while PGD converges to a constant noise level.

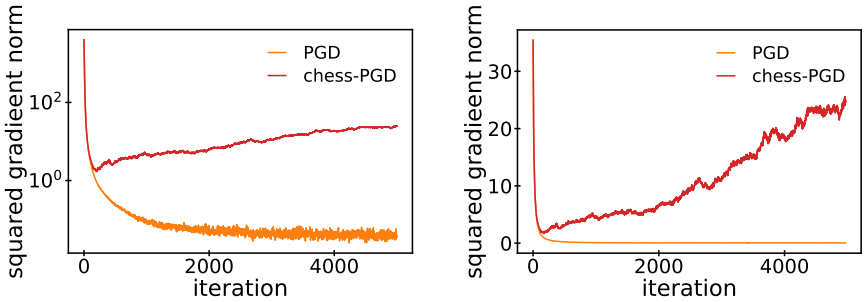



Figure 4: Comparison of PGD and Chess-PGD under the fixed stepsize, $\gamma = 0.02$. Y axis in the log scale on the left, and in the normal scale on the right.

# B Factorization Matrices

As discussed in Section 2, Denisov et al. [10] propose finding useful factorizations for DP training by solving the problem

$$\min_{\mathbf{B},\mathbf{C}} \|\mathbf{B}\|_F^2 \quad \text{such that} \quad \mathbf{BC} = \mathbf{A}, \quad \text{sens}(\mathbf{C}) = 1. \tag{17}$$

As we discuss in Section 5, based on our convergence rates in Section 4, we propose the following modified objective:

$$\min_{\mathbf{B},\mathbf{C}} \|\Lambda_\tau \mathbf{B}\|_F^2 \quad \text{such that } \mathbf{BC} = \mathbf{A}, \ \text{sens}(\mathbf{C}) = 1. \tag{18}$$

The matrix $\Lambda_\tau = [\lambda_{tj}]_{t,j=1,\dots,T}$ is defined as follows:

$$\lambda_{tj} = \begin{cases} \frac{1}{\sqrt{\tau}} & j = t, \quad t \neq 0 \bmod \tau \\ -\frac{1}{\sqrt{\tau}} & j = \lfloor \frac{t}{\tau} \rfloor \tau, \quad t \neq 0 \bmod \tau, t > \tau \\ 1 & j = t, \quad t = 0 \bmod \tau \\ -1 & j = t - \tau, \quad t = 0 \bmod \tau, t > \tau \end{cases}$$

For all the other indices, $\lambda_{tj} = 0$. In Figure 5 we give an example of such a matrix for $T = 12$ and $\tau = 3$.

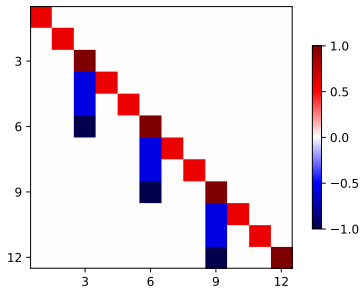
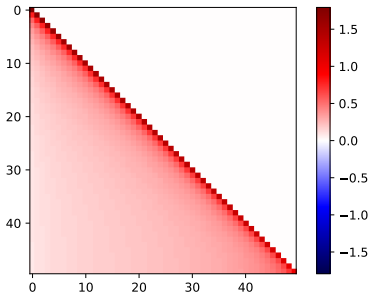
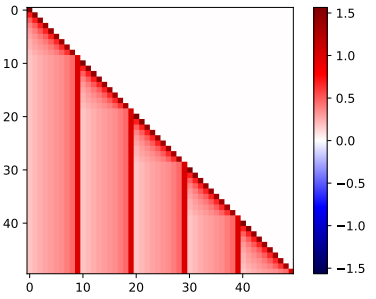



Figure 5: Elements of $\Lambda_\tau$ for $T = 12$, and $\tau = 3$.

To illustrate how the parameter $\tau$ affects the solution to the objective problem, we plot numerically computed approximate minimizers to (17) and (18) in Figure 6(a) and Figure 6(b), respectively. Specifically, we plot the matrix $\mathbf{B}$, and let $\mathbf{B}_{\text{MF}}$ denote the solution to (17) and $\mathbf{B}_{\text{MF}^+}$ denote the solution to (18). We can clearly see that for the latter, the parameter $\tau$ enforces a block-like structure such that the bands of correlation are at regular intervals of length $\tau$.




(a) $\mathbf{B}_{\text{MF}}$, $T = 50$




(b) $\mathbf{B}_{\text{MF}^+}$, $T = 50$, $\tau = 10$

## C Proofs of Main Results

We analyse the algorithm with general $\mathbf{B}$ that has the following iterates:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_t)^\top \mathbf{Z}) \quad t \geq 1 \tag{19}$$

where $\mathbf{b}_0 = 0$. We define $\mathbf{v}_t = (\mathbf{b}_{t+1} - \mathbf{b}_t)^\top \mathbf{Z}$ for $t \geq 0$, so that

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t) - \gamma \mathbf{v}_t$$

For the analysis, we define a virtual sequence with restarts (14), where we do restarts every $\tau$ iterations. Formally, we define virtual iterates $\{\tilde{\mathbf{x}}_t\}_{t=0}^T$ as follows:

$$\begin{aligned}
&\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) && \text{if } t+1 \neq 0 \bmod \tau \\
&\tilde{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1} && \text{if } t+1 = 0 \bmod \tau \text{ (restart iterations)}
\end{aligned}$$

This means that $\tilde{\mathbf{x}}_{k\tau} = \mathbf{x}_{k\tau}$, for any nonnegative integer $k$.

**Useful facts about this sequence.**

- The closest restart iteration to $t$ is equal to $\lfloor \frac{t}{\tau} \rfloor \tau$.
- For $t < \tau$ we have

$$\tilde{\mathbf{x}}_t - \mathbf{x}_t = \gamma \mathbf{b}_t^\top \mathbf{Z}$$

- For restart iterations $t+1 = \tau$,

$$\tilde{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1} = 0$$

- For the next iteration just after restart $t+1 = \tau + 1$

$$\tilde{\mathbf{x}}_{\tau+1} - \mathbf{x}_{\tau+1} = (\tilde{\mathbf{x}}_\tau - \gamma \nabla f(\mathbf{x}_\tau)) - (\mathbf{x}_\tau - \gamma \nabla f(\mathbf{x}_\tau) - \gamma \mathbf{v}_\tau) = \gamma \mathbf{v}_\tau = \gamma (\mathbf{b}_{\tau+1} - \mathbf{b}_\tau)^\top \mathbf{Z}$$

- Thus, for arbitrary $t$,

$$\tilde{\mathbf{x}}_t - \mathbf{x}_t = \gamma (\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z} \tag{20}$$

  (and if $t = 0 \bmod \tau$, then the term cancels and we get $\tilde{\mathbf{x}}_t - \mathbf{x}_t = 0$), we assume that $\mathbf{b}_0 = \mathbf{0}$.
- We can re-write the restart iterations for $t+1 = 0 \bmod \tau$

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1} &= \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t) - \gamma (\mathbf{b}_{t+1} - \mathbf{b}_t)^\top \mathbf{Z} \\
&= \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) - \gamma (\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z} - \gamma (\mathbf{b}_{t+1} - \mathbf{b}_t)^\top \mathbf{Z} \\
&= \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) - \gamma (\mathbf{b}_{t+1} - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z}
\end{aligned}$$

  Equivalently, for $t+1 = 0 \bmod \tau$,

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t) - \gamma (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z}. \tag{21}$$

### C.1 Assumptions and Useful Inequalities

This section contains assumptions and inequalities that will be used throughout the proof. First, recall that in Assumption 4.2, we assume that $f$ is differentiable and $L$-smooth, so that

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|. \tag{22}$$

In some settings, we will also assume convexity, so that

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad f(\mathbf{x}) - f(\mathbf{y}) \leq \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle. \tag{23}$$

We will also make use of the following facts about the geometry of vectors in $\mathbb{R}^d$.

**Lemma C.1.** *For any finite set of vectors $\{\mathbf{a}_i\}_{i=1}^n \subset \mathbb{R}^d$,*

$$\left\| \sum_{i=1}^n \mathbf{a}_i \right\|^2 \leq n \sum_{i=1}^n \|\mathbf{a}_i\|^2. \tag{24}$$

**Lemma C.2.** *For any two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and for all $\alpha > 0$,*

$$2\langle \mathbf{a}, \mathbf{b} \rangle \leq \alpha \|\mathbf{a}\|^2 + \alpha^{-1} \|\mathbf{b}\|^2. \tag{25}$$

### C.2 Proof for Non-convex Functions

**Iterations without restarts.** If $t$ is such that $t \neq -1 \bmod \tau$, where $k$ is some integer number, then between iteration $t$ and $t+1$ no restart of virtual sequence happens and thus $\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t)$. We follow closely standard perturbed iterate analysis [28, 43]. By $L$-smoothness of $f$

$$
\begin{aligned}
f(\tilde{\mathbf{x}}_{t+1}) &\leq f(\tilde{\mathbf{x}}_t) - \gamma \langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t) \rangle + \frac{L\gamma^2}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 \\
&\leq f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 - \frac{\gamma}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma L^2}{2} \left\| \mathbf{x}_t - \tilde{\mathbf{x}}_t \right\|^2 \\
&\overset{(20)}{\leq} f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 - \frac{\gamma}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma^3 L^2}{2} \left\| (\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z} \right\|^2 \qquad (26)
\end{aligned}
$$

where on the second line we used that $-2\langle \mathbf{a}, \mathbf{b} \rangle = -\left\| \mathbf{a} \right\|^2 - \left\| \mathbf{b} \right\|^2 + \left\| \mathbf{a} - \mathbf{b} \right\|^2$ for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$.

**Iterations with restarts.** Restart happens between iteration $t$ and $t+1$ if $t = -1 \bmod \tau$. In this case, the analysis is more involved. By $L$-smoothness and using update rule (21)

$$
\begin{aligned}
f(\tilde{\mathbf{x}}_{t+1}) &\leq f(\tilde{\mathbf{x}}_t) - \gamma \langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \rangle && (27) \\
&\quad + \frac{L}{2} \gamma^2 \left\| \nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \right\|^2 && (28) \\
&\overset{(24)}{\leq} f(\tilde{\mathbf{x}}_t) - \underbrace{\gamma \langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t) \rangle}_{:= T_1} - \underbrace{\gamma \langle \nabla f(\tilde{\mathbf{x}}_t), (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \rangle}_{:= T_2} && (29) \\
&\quad + L\gamma^2 \left\| \nabla f(\mathbf{x}_t) \right\|^2 + L\gamma^2 \left\| (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \right\|^2 && (30)
\end{aligned}
$$

We estimate separately the second and the third terms

$$
\begin{aligned}
T_1 &= -\frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 - \frac{\gamma}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t) \right\|^2 \\
&\overset{(22)}{\leq} -\frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 - \frac{\gamma}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma L^2}{2} \left\| \tilde{\mathbf{x}}_t - \mathbf{x}_t \right\|^2 \\
&\overset{(20)}{\leq} -\frac{\gamma}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 - \frac{\gamma}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma^3 L^2}{2} \left\| (\mathbf{b}_t - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z} \right\|^2
\end{aligned}
$$

The third term,

$$
\begin{aligned}
T_2 &= -\langle \nabla f(\tilde{\mathbf{x}}_t), \gamma (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \rangle \\
&\overset{(25),\ \alpha = \frac{1}{8L}}{\leq} \frac{1}{16L} \left\| \nabla f(\tilde{\mathbf{x}}_t) \right\|^2 + 4L\gamma^2 \left\| (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top \mathbf{Z} \right\|^2
\end{aligned}
$$

It is left to deal with the norm of the gradient $\frac{1}{16L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2$. Using that $\tau = \frac{1}{L\gamma}$, and thus $\frac{1}{16L\tau} = \frac{\gamma}{16}$ we have

$$\begin{aligned}
\frac{1}{16L}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 &= \frac{\gamma}{16}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 \\
&\overset{(24),(22)}{\leq} \frac{\gamma}{8}\sum_{i=0}^{\tau-1} L^2\left\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-i}\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2 \\
&\overset{(26)}{\leq} \frac{\gamma}{8}\sum_{i=1}^{\tau-1}\gamma^2 L^2\left\|\sum_{j=t-i}^{t-1}\nabla f(\mathbf{x}_j)\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2 \\
&\overset{(24)}{\leq} \frac{\gamma^3 L^2}{8}\sum_{i=1}^{\tau-1}\tau\sum_{j=t-i}^{t-1}\left\|\nabla f(\mathbf{x}_j)\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2 \\
&\leq \frac{\gamma^3 L^2\tau^2}{8}\sum_{i=1}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-i})\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2 \\
&\overset{\tau=\frac{1}{\gamma L}}{\leq} \frac{\gamma}{8}\sum_{i=1}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-i})\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2
\end{aligned}$$

Putting back our calculations of $T_1$ and $T_2$ into (30), and setting $\gamma \leq \frac{1}{4L}$ in order to estimate that $L\gamma^2\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{\gamma}{4}\left\|\nabla f(\mathbf{x}_t)\right\|^2$

$$\begin{aligned}
f(\tilde{\mathbf{x}}_{t+1}) \leq f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{2}\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 - \frac{\gamma}{4}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma^3 L^2}{2}\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2 \\
+ 5L\gamma^2\left\|(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2 + \frac{\gamma}{8}\sum_{i=1}^{\tau-1}\left\|\nabla f(\mathbf{x}_{t-i})\right\|^2 + \frac{\gamma}{8}\sum_{i=0}^{\tau-1}\left\|\nabla f(\tilde{\mathbf{x}}_{t-i})\right\|^2
\end{aligned} \tag{31}$$

**Combining iterations with and without restarts.** It is left to average equations (26) and (31) over all iterations $0 \leq t \leq T$. We denote $\mathcal{T}_1$ is the set of indices without restarts, and $\mathcal{T}_2$ are restarts indices.

$$\begin{aligned}
&\sum_{t\in\mathcal{T}_1}\frac{\gamma}{8}\left(\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 + \left\|\nabla f(\mathbf{x}_t)\right\|^2\right) + \sum_{t\in\mathcal{T}_2}\frac{\gamma}{8}\left(\left\|\nabla f(\tilde{\mathbf{x}}_t)\right\|^2 + \left\|\nabla f(\mathbf{x}_t)\right\|^2\right) \\
&\qquad \leq (f(\mathbf{x}_0) - f^\star) + \frac{\gamma^3 L^2}{2}\sum_{t=1}^{T}\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2 + 5L\gamma^2\sum_{t\in\mathcal{T}_1}\left\|(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2
\end{aligned}$$

Dividing by $\frac{\gamma(T+1)}{8}$, we get

$$\begin{aligned}
\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{8(f(\mathbf{x}_0) - f^\star)}{\gamma(T+1)} + \frac{4\gamma^2 L^2}{T+1}\sum_{t=1}^{T}\mathbb{E}\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2 \\
+ \frac{40L\gamma}{T+1}\sum_{k=1}^{\lfloor\frac{T}{\tau}\rfloor}\mathbb{E}\left\|(\mathbf{b}_{k\tau} - \mathbf{b}_{(k-1)\tau})^\top\mathbf{Z}\right\|^2
\end{aligned}$$

which completes the proof.

### C.3 Proof for Convex Functions

Our proof for convex functions follows the same pattern as for non-convex: we consider separately iterations with and without restarts of the virtual sequence (14). However, summing up these two cases is the most involved part of the proof in the convex case, and it is different from the non-convex case.

We will use the following fact in our proof.

**Lemma C.3.** *If function $f$ is convex* (23)*, $L$-smooth* (22)*, and has a finite minimizer $x^*$, then*

$$\|\nabla f(\mathbf{x})\|^2 \leq 2L\left(f(\mathbf{x}) - f^\star\right). \tag{32}$$

**Iterations without restarts.** Using (14), i.e. that $\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma\nabla f(\mathbf{x}_t)$, for some point $\mathbf{x}^\star$ that satisfies $\nabla f(\mathbf{x}^\star) = 0$,

$$\begin{aligned}\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 &= \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - 2\gamma\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + 2\gamma\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \tilde{\mathbf{x}}_t\rangle \\ &\overset{(32),(23)}{\leq} \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - 2\gamma(1 - L\gamma)\left(f(\mathbf{x}_t) - f^\star\right) + 2\gamma\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \tilde{\mathbf{x}}_t\rangle\end{aligned}$$

We estimate the last term separately

$$2\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \tilde{\mathbf{x}}_t\rangle \overset{(25),\alpha=2L}{\leq} \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2 + 2L\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|^2 \overset{(32)}{\leq} \left(f(\mathbf{x}_t) - f^\star\right) + 2L\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|^2$$

Thus,

$$\begin{aligned}\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 &\leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - \gamma(1 - 2L\gamma)(f(\mathbf{x}_t) - f^\star) + 2L\gamma\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|^2 \\ &\overset{\gamma<\frac{1}{4L},(20)}{\leq} \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - \frac{\gamma}{2}(f(\mathbf{x}_t) - f^\star) + 2L\gamma^3\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2\end{aligned} \tag{33}$$

**For the iterations with restarts.** This means that $t + 1 = k\tau$. Using (21),

$$\begin{aligned}\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 &= \left\|\tilde{\mathbf{x}}_t - \mathbf{x}^\star - \gamma\nabla f(\mathbf{x}_t) - \gamma(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2 \\ &= \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - 2\gamma\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle - 2\gamma\langle(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle \\ &\quad + \gamma^2\left\|\nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2\end{aligned}$$

We estimate the second term same as in the case without restarts:

$$\begin{aligned}-2\gamma\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle &= -2\gamma\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle - 2\gamma\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{x}}_t - \mathbf{x}_t\rangle \\ &\leq -\gamma(f(\mathbf{x}_t) - f^\star) + 2L\gamma\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|^2\end{aligned}$$

For the last term,

$$\begin{aligned}\gamma^2\left\|\nabla f(\mathbf{x}_t) + (\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2 &\overset{(24)}{\leq} 2\gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + 2\gamma^2\left\|(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2 \\ &\overset{(32)}{\leq} 4L\gamma^2(f(\mathbf{x}_t) - f^\star) + 2\gamma^2\left\|(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2\end{aligned}$$

Thus with $\gamma \leq \frac{1}{8L}$,

$$\begin{aligned}\frac{\gamma}{2}(f(\mathbf{x}_t) - f^\star) &\leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 - \|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 + 2L\gamma^3\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2 \\ &\quad + 2\gamma^2\left\|(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}\right\|^2 - 2\gamma\langle(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle\end{aligned} \tag{34}$$

**Combining iterations with and without restarts.** Summing up (33) and (34) for all $0 \leq t \leq T$,

$$\begin{aligned}\frac{\gamma}{2}\sum_{t=0}^{T}(f(\mathbf{x}_t) - f^\star) &\leq \|\tilde{\mathbf{x}}_0 - \mathbf{x}^\star\|^2 - \|\tilde{\mathbf{x}}_{T+1} - \mathbf{x}^\star\|^2 + 2L\gamma^3\sum_{t=0}^{T}\left\|(\mathbf{b}_t - \mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2 \\ &\quad + 2\gamma^2\sum_{k=1}^{\lfloor\frac{T}{\tau}\rfloor}\mathbb{E}\left\|(\mathbf{b}_{k\tau} - \mathbf{b}_{(k-1)\tau})^\top\mathbf{Z}\right\|^2 \underbrace{-2\gamma\sum_{k=1}^{\lfloor\frac{T}{\tau}\rfloor}\langle(\mathbf{b}_{k\tau} - \mathbf{b}_{(k-1)\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_{k\tau-1} - \mathbf{x}^\star\rangle}_{:=S_1}\end{aligned} \tag{35}$$

We now separately estimate the last sum $S_1$. We first divide it in pairs of two consecutive terms, and sum each pair separately. Lets denote $t = k\tau - 1$ for some $k$. Sum of two consecutive terms with indexes $t$ and $t - \tau$ is equal to

$$\begin{aligned}&-2\gamma\langle(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle - 2\gamma\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_{t-\tau} - \mathbf{x}^\star\rangle \\ =&-2\gamma\langle(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle - 2\gamma\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle \\ &\quad - 2\gamma\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_{t-\tau} - \tilde{\mathbf{x}}_t\rangle \\ =&-2\gamma\langle(\mathbf{b}_{t+1} - \mathbf{b}_{t+1-2\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle - 2\gamma\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top\mathbf{Z}, \tilde{\mathbf{x}}_{t-\tau} - \tilde{\mathbf{x}}_t\rangle\end{aligned}$$

Using update rules (14), it holds that $\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-\tau} - \gamma \sum_{j=t-\tau}^{t-1} \nabla f(\mathbf{x}_j) - \gamma(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}$, and thus

$$
\begin{aligned}
&-2\gamma\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}, \tilde{\mathbf{x}}_{t-\tau} - \tilde{\mathbf{x}}_t\rangle \\
&\qquad = -2\gamma^2\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}, \sum_{j=t-\tau}^{t-1} \nabla f(\mathbf{x}_j) + (\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}\rangle \\
&\qquad = \sum_{j=t-\tau}^{t-1} -2\gamma^2\langle(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}, \nabla f(\mathbf{x}_j)\rangle - 2\gamma^2 \left\|(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}\right\|^2 \\
&\qquad \overset{(25)}{\leq} \gamma^2\alpha\tau \left\|(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}\right\|^2 + \gamma^2\alpha^{-1} \sum_{j=t-\tau}^{t-1} \|\nabla f(\mathbf{x}_t)\|^2 \\
&\qquad\qquad - 2\gamma^2 \left\|(\mathbf{b}_{t+1-\tau} - \mathbf{b}_{t+1-2\tau})^\top \mathbf{Z}\right\|^2 \\
&\qquad \overset{\alpha=\frac{2}{\tau}}{\leq} \frac{\gamma^2\tau}{2} \sum_{j=t-\tau}^{t-1} \|\nabla f(\mathbf{x}_t)\|^2
\end{aligned}
$$

Using these calculations, our original sum $S_1$ can be simplified as

$$
S_1 \leq -2\gamma \sum_{k=1}^{\lfloor \frac{T}{2\tau} \rfloor} \langle(\mathbf{b}_{k\cdot 2\tau} - \mathbf{b}_{(k-1)2\tau})^\top \mathbf{Z}, \tilde{\mathbf{x}}_{k\cdot 2\tau-1} - \mathbf{x}^\star\rangle + \frac{\gamma^2\tau}{2} \sum_{t=0}^{\lfloor \frac{T}{\tau} \rfloor \tau - 2} \|\nabla f(\mathbf{x}_t)\|^2
$$

We reduced the sum of $\lfloor \frac{T}{\tau} \rfloor$ elements twice to the sum of the $\lfloor \frac{T}{2\tau} \rfloor$ elements. Continuing in similar way, we will need to have $\log_2\left(\lfloor \frac{T}{\tau} \rfloor\right)$ times until we reduce the original sum to just one element. Thus,

$$
\begin{aligned}
S_1 &\leq -2\gamma\langle \mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau}^\top \mathbf{Z}, \tilde{\mathbf{x}}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{x}^\star\rangle + \frac{\gamma^2\tau}{2} \log_2\left(\left\lfloor \frac{T}{\tau} \right\rfloor\right) \sum_{t=0}^{\lfloor \frac{T}{\tau} \rfloor \tau - 2} \|\nabla f(\mathbf{x}_t)\|^2 \\
&\overset{(25),\alpha=2}{\leq} \frac{1}{3}\left\|\tilde{\mathbf{x}}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{x}^\star\right\|^2 + 3\gamma^2 \left\|\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau}^\top \mathbf{Z}\right\|^2 + \frac{\gamma^2\tau}{2} \log_2\left(\left\lfloor \frac{T}{\tau} \right\rfloor\right) \sum_{t=0}^{\lfloor \frac{T}{\tau} \rfloor \tau - 2} \|\nabla f(\mathbf{x}_t)\|^2
\end{aligned}
$$

We further transform the first term using the update rule (14)

$$
\tilde{\mathbf{x}}_{T+1} = \tilde{\mathbf{x}}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \gamma \sum_{j=\lfloor \frac{T}{\tau} \rfloor \tau}^{T} \nabla f(\mathbf{x}_j) = \tilde{\mathbf{x}}_{\lfloor \frac{T}{\tau} \rfloor \tau - 1} - \gamma \sum_{j=\lfloor \frac{T}{\tau} \rfloor \tau - 1}^{T} \nabla f(\mathbf{x}_j) - \gamma \left(\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{b}_{(\lfloor \frac{T}{\tau} \rfloor - 1)\tau}\right)^\top \mathbf{Z}
$$

Thus,

$$
\frac{1}{3}\left\|\tilde{\mathbf{x}}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{x}^\star\right\|^2 \leq \|\tilde{\mathbf{x}}_{T+1} - \mathbf{x}^\star\|^2 + \gamma^2\tau \sum_{j=\lfloor \frac{T}{\tau} \rfloor \tau - 1}^{T} \|\nabla f(\mathbf{x}_j)\|^2 + \gamma^2 \left\|\left(\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{b}_{(\lfloor \frac{T}{\tau} \rfloor - 1)\tau}\right)^\top \mathbf{Z}\right\|^2
$$

And thus,

$$
\begin{aligned}
S_1 \leq \|\tilde{\mathbf{x}}_{T+1} - \mathbf{x}^\star\|^2 &+ 3\gamma^2 \left\|\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau}^\top \mathbf{Z}\right\|^2 + \gamma^2\tau \log_2\left(\left\lfloor \frac{T}{\tau} \right\rfloor\right) \sum_{t=0}^{T} \|\nabla f(\mathbf{x}_t)\|^2 \\
&+ \gamma^2 \left\|\left(\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau} - \mathbf{b}_{(\lfloor \frac{T}{\tau} \rfloor - 1)\tau}\right)^\top \mathbf{Z}\right\|^2
\end{aligned}
$$

Choosing $\tau = \frac{1}{8L\gamma \log_2(T)}$ ensures that $\gamma^2\tau \log_2\left(\lfloor \frac{T}{\tau} \rfloor\right) \leq \frac{\gamma}{8L}$. Putting these calculations back into (35), we get that

$$
\begin{aligned}
\frac{\gamma}{2}\sum_{t=0}^{T}(f(\mathbf{x}_t) - f^\star) \leq \|\tilde{\mathbf{x}}_0 - \mathbf{x}^\star\|^2 &+ 3L\gamma^3 \sum_{t=0}^{T} \left\|(\mathbf{b}_{t-1} - \mathbf{b}_{\lfloor \frac{t}{\tau} \rfloor \tau})^\top \mathbf{Z}\right\|^2 \\
&+ 3\gamma^2 \sum_{k=1}^{\lfloor \frac{T}{\tau} \rfloor} \mathbb{E}\left\|(\mathbf{b}_{k\tau} - \mathbf{b}_{(k-1)\tau})^\top \mathbf{Z}\right\|^2 + \frac{\gamma}{8L} \sum_{t=0}^{T} \|\nabla f(\mathbf{x}_t)\|^2 + 3\gamma^2 \left\|\mathbf{b}_{\lfloor \frac{T}{\tau} \rfloor \tau}^\top \mathbf{Z}\right\|^2
\end{aligned}
$$

Using (32), we can further simplify

$$\frac{\gamma}{4}\sum_{t=0}^{T}(f(\mathbf{x}_t)-f^\star)\le 3\gamma^2\Big(L\gamma\sum_{t=0}^{T}\left\|(\mathbf{b}_t-\mathbf{b}_{\lfloor\frac{t}{\tau}\rfloor\tau})^\top\mathbf{Z}\right\|^2+\sum_{k=1}^{\lfloor\frac{T}{\tau}\rfloor}\mathbb{E}\left\|(\mathbf{b}_{k\tau}-\mathbf{b}_{(k-1)\tau})^\top\mathbf{Z}\right\|^2 + \left\|\mathbf{b}_{\lfloor\frac{T}{\tau}\rfloor\tau}^\top\mathbf{Z}\right\|^2\Big)+\left\|\mathbf{x}_0-\mathbf{x}^\star\right\|^2$$

# D Convergence of Anti-PGD

Here we discuss the convergence of the Anti-PGD method, introduced in Example 3.2.

Since $\tilde{\mathbf{x}}_{t+1}=\tilde{\mathbf{x}}_t-\gamma\nabla f(\mathbf{x}_t)$, for some point $\mathbf{x}^\star$ that satisfies $\nabla f(\mathbf{x}^\star)=0$,

$$\begin{aligned}\left\|\tilde{\mathbf{x}}_{t+1}-\mathbf{x}^\star\right\|^2&=\left\|\tilde{\mathbf{x}}_t-\mathbf{x}^\star\right\|^2-2\gamma\langle\nabla f(\mathbf{x}_t),\mathbf{x}_t-\mathbf{x}^\star\rangle+\gamma^2\left\|\nabla f(\mathbf{x}_t)\right\|^2+2\gamma\langle\nabla f(\mathbf{x}_t),\mathbf{x}_t-\tilde{\mathbf{x}}_t\rangle\\&\overset{(32),(23)}{\le}\left\|\tilde{\mathbf{x}}_t-\mathbf{x}^\star\right\|^2-2\gamma(1-L\gamma)\left(f(\mathbf{x}_t)-f^\star\right)+2\gamma\langle\nabla f(\mathbf{x}_t),\mathbf{x}_t-\tilde{\mathbf{x}}_t\rangle\end{aligned}$$

We estimate the last term separately

$$2\langle\nabla f(\mathbf{x}_t),\mathbf{x}_t-\tilde{\mathbf{x}}_t\rangle\overset{(25),\alpha=2L}{\le}\frac{1}{2L}\left\|\nabla f(\mathbf{x}_t)\right\|^2+2L\left\|\mathbf{x}_t-\tilde{\mathbf{x}}_t\right\|^2\overset{(32)}{\le}\left(f(\mathbf{x}_t)-f^\star\right)+2L\left\|\mathbf{x}_t-\tilde{\mathbf{x}}_t\right\|^2$$

Thus,

$$\begin{aligned}\left\|\tilde{\mathbf{x}}_{t+1}-\mathbf{x}^\star\right\|^2&\le\left\|\tilde{\mathbf{x}}_t-\mathbf{x}^\star\right\|^2-\gamma(1-2L\gamma)(f(\mathbf{x}_t)-f^\star)+2L\gamma\left\|\mathbf{x}_t-\tilde{\mathbf{x}}_t\right\|^2\\&\overset{\gamma<\frac{1}{4L},(20)}{\le}\left\|\tilde{\mathbf{x}}_t-\mathbf{x}^\star\right\|^2-\frac{\gamma}{2}(f(\mathbf{x}_t)-f^\star)+2L\gamma\left\|\mathbf{x}_t-\tilde{\mathbf{x}}_t\right\|^2\end{aligned}$$

# E Noise Lower Bound

We consider function $f(\mathbf{x})=\frac{L}{2}\left\|\mathbf{x}\right\|^2$ that is convex and $L$-smooth, and we are running algorithm (8) with constant stepsize $\gamma$, and we consider the two cases of $\mathbf{B}=\mathbf{S}$ and $\mathbf{B}=\mathbf{I}$.

## E.1 PGD

This corresponds to Example 3.1. We will prove the lower bound on the noise term under the condition that $T$ is large enough, i.e. $T\ge\frac{\log 2}{\eta L}$.

Since $\nabla f(\mathbf{x})=L\mathbf{x}$, the algorithm (8) takes a form

$$\mathbf{x}_{t+1}=(1-\gamma L)\mathbf{x}_t-\gamma\mathbf{z}_{t+1}$$

Thus, since $\mathbf{x}^\star=0$,

$$\begin{aligned}\mathbb{E}\left\|\mathbf{x}_{t+1}\right\|^2&=\mathbb{E}\left\|(1-\gamma L)\mathbf{x}_t-\gamma\mathbf{z}_{t+1}\right\|^2=(1-\gamma L)^2\left\|\mathbf{x}_t\right\|^2+\gamma^2\sigma^2\\&=(1-\gamma L)^{2(t+1)}\left\|\mathbf{x}_0\right\|^2+\gamma^2\sigma^2\sum_{j=0}^{t}(1-\gamma L)^{2j}\end{aligned}$$

due to the unbiasedness and independence of $\mathbf{z}_t$. We can exactly calculate the sum of this geometric series

$$\sum_{j=0}^{T-1}(1-\gamma L)^{2j}=\frac{1-(1-\gamma L)^{2T}}{1-(1-\gamma L)^2}=\frac{1-(1-2\gamma L)^{2T}}{2\gamma L-\gamma^2L^2}\ge\frac{1}{4\gamma L}$$

where at the last step we used that $\gamma^2L^2>0$ and that $T\ge\frac{\log 2}{\gamma L}$.

And thus the function values are larger than

$$f(\mathbf{x}_T)-f^\star=\frac{L}{2}\left\|\mathbf{x}_T\right\|^2\ge\frac{L}{2}(1-\gamma L)^{2(t+1)}\left\|\mathbf{x}_0\right\|^2+\frac{1}{8}\gamma\sigma^2$$

This shows that the noise term in (11) cannot be improved.

### E.2 Anti-PGD

This corresponds to Example 3.2. Since $\nabla f(\mathbf{x}) = L\mathbf{x}$, the algorithm (8) for Anti-PGD noise takes a form

$$\begin{aligned}\mathbf{x}_{t+1} &= (1-\gamma L)\mathbf{x}_t - \gamma \mathbf{z}_{t+1} + \gamma \mathbf{z}_t \\ &= (1-\gamma L)^2 \mathbf{x}_{t-1} - (1-\gamma L)\gamma \mathbf{z}_t + (1-\gamma L)\gamma \mathbf{z}_{t-1} - \gamma \mathbf{z}_{t+1} + \gamma \mathbf{z}_t \\ &= (1-\gamma L)^2 \mathbf{x}_{t-1} + (1-\gamma L)\gamma \mathbf{z}_{t-1} - \gamma \mathbf{z}_{t+1} + \gamma^2 L \mathbf{z}_t \\ &= (1-\gamma L)^{t+1}\mathbf{x}_0 + (1-\gamma L)^t \gamma \mathbf{z}_1 + \gamma^2 L \sum_{j=1}^{t-1} (1-\gamma L)^{t-j}\mathbf{z}_{j+1} - \gamma \mathbf{z}_{t+1}\end{aligned}$$

Thus,

$$\mathbb{E}\left\|\mathbf{x}_T\right\|^2 = (1-\gamma L)^{2T}\left\|\mathbf{x}_0\right\|^2 + (1-\gamma L)^{2(T-1)}\gamma^2 \mathbb{E}\left\|\mathbf{z}_1\right\|^2 + \gamma^4 L^2 \sum_{j=1}^{T-2}(1-\gamma L)^{2(T-1-j)}\mathbb{E}\left\|\mathbf{z}_{j+1}\right\|^2$$

$$+ \gamma^2 \mathbb{E}\left\|\mathbf{z}_{t+1}\right\|^2 \geq (1-\gamma L)^{2T}\left\|\mathbf{x}_0\right\|^2 + \gamma^2\sigma^2$$

Thus the function values are larger than

$$f(\mathbf{x}_T) - f^\star = \frac{L}{2}\left\|\mathbf{x}_T\right\|^2 \geq (1-\gamma L)^{2T}\left\|\mathbf{x}_0\right\|^2 + \frac{L}{2}\gamma^2\sigma^2$$

This proves that the noise term in (12) cannot be improved.

### E.3 Virtual Sequence for PGD

In this section we show that for the PGD algorithm, virtual sequences $\tilde{\mathbf{x}}_t$ that are defined in (13) cannot give a tight convergence result.

Since $\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \nabla f(\mathbf{x}_t)$, and $\nabla f(\mathbf{x}_t) = L\mathbf{x}_t$ we get

$$\tilde{\mathbf{x}}_{t+1} = (1-\gamma L)\tilde{\mathbf{x}}_t + \gamma L(\tilde{\mathbf{x}}_t - \mathbf{x}_t) = (1-\gamma L)\tilde{\mathbf{x}}_t - \gamma^2 L \sum_{j=0}^{t}\mathbf{z}_j$$

where the last equality is since $\tilde{\mathbf{x}}_t - \mathbf{x}_t = -\gamma\sum_{j=1}^{t}\mathbf{z}_j$. Unrolling,

$$\tilde{\mathbf{x}}_{t+1} = (1-\gamma L)\tilde{\mathbf{x}}_t + \gamma L(\tilde{\mathbf{x}}_t - \mathbf{x}_t) = (1-\gamma L)^{t+1}\tilde{\mathbf{x}}_0 - \gamma^2 L \sum_{j=1}^{t}\mathbf{z}_j \sum_{i=0}^{j}(1-\gamma L)^i$$

Thus the norm

$$\mathbb{E}\left\|\tilde{\mathbf{x}}_T\right\|^2 = (1-\gamma L)^{2T}\left\|\mathbf{x}_0\right\|^2 + \gamma^4 L^2 \sum_{j=1}^{T-1}\left[\sum_{i=0}^{j}(1-\gamma L)^i\right]^2 \sigma^2$$

We can calculate exactly the inner sum as

$$\sum_{i=0}^{j}(1-\gamma L)^i = \frac{1-(1-\gamma L)^j}{\gamma L}$$

and thus

$$\mathbb{E}\left\|\tilde{\mathbf{x}}_T\right\|^2 = (1-\gamma L)^{2T}\left\|\mathbf{x}_0\right\|^2 + \gamma^2 \sum_{j=1}^{T-1}\left[1-(1-\gamma L)^j\right]^2\sigma^2 \geq \gamma^2 \sum_{j=\frac{T}{2}}^{T-1}\left[1-2(1-\gamma L)^j\right]\sigma^2$$

It is left to note that for $T$ sufficiently large, $T \geq 2\frac{\log 4}{\gamma L}$, it holds that $(1-\gamma L)^{T/2} \leq \frac{1}{4}$ and thus $\left[1-2(1-\gamma L)^j\right] \geq \frac{1}{2}$. Using this, we arrive

$$\mathbb{E}\left\|\tilde{\mathbf{x}}_T\right\|^2 \geq \gamma^2\sigma^2\frac{T}{4}$$

and this the function value $f(\tilde{\mathbf{x}}_T) \geq L\gamma^2\sigma^2\frac{T}{8}$.

## F Difficulties in Deriving a Unified Analysis

In this section we explain the difficulties in unifying theoretical analysis using existing proof techniques described in the main text. In particular analysis through the real iterates $\mathbf{x}_t$ can give good convergence guarantees only for PGD, but not Anti-PGD, and vise versa, analysis through the virtual iterates $\tilde{\mathbf{x}}_t$ can give a good convergence guarantee for Anti-PGD but not for PGD.

Directly analyzing Anti-PGD using the actual iterates $\mathbf{x}_t$ of (7), we only get a convergence rate of

$$\sum_{t=0}^{T} \frac{\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right]}{T+1} \leq \mathcal{O}\left(\frac{\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|^2}{\gamma T} + \gamma\sigma^2\right).$$

Note that this is strictly worse than the Anti-PGD rate in (12). While we do not see any fundamental limit to analysing Anti-PGD directly through its iterates $\mathbf{x}_t$, we do not know of how to do so in a way that recovers the rate in (12).

On the other hand, applying the perturbed iterate analysis (via the virtual sequence $\tilde{\mathbf{x}}_t$ produced by (7) when $\mathbf{Z} = \mathbf{0}$) to PGD, we only get a convergence rate of

$$\sum_{t=0}^{T} \frac{\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right]}{T+1} \leq \mathcal{O}\left(\frac{\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|^2}{\gamma T} + LT\gamma^2\sigma^2\right).$$

This rate is strictly worse than the rate derived through a virtual sequence in (11) when $\gamma > 1/LT$. As we detail in Appendix E, this bound is actually a tight upper bound for the convergence of the virtual sequence $f(\tilde{\mathbf{x}}_t)$. However, the real sequence $\mathbf{x}_t$ converges faster than this according to (11). In short, while one can use the virtual sequence $\tilde{\mathbf{x}}_t$ to effectively analyze anti-correlated noise, such techniques do not directly yield a tight analysis of PGD.

## G Applying Theorem 4.7 to special cases

**PGD.** In this case, $\mathbf{B} = \mathbf{S}$ (Example 2.1), so if $i - j \leq \tau$ then $\left\|\mathbf{b}_i - \mathbf{b}_j\right\|^2 \leq \tau$. The noise term in the convergence rate of Theorem 4.7 is therefore upper bounded by

$$\frac{\sigma^2}{TL\tau}\left[\frac{1}{\tau}\sum_{t=1}^{T}\tau + \sum_{\substack{1\leq t\leq T\\ t=0 \bmod \tau}}\tau + T\right] = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{L\tau}\right) = \tilde{\mathcal{O}}\left(\gamma\sigma^2\right)$$

This matches the tight convergence rate in Proposition 4.4.

**Anti-PGD.** Since $\mathbf{B} = \mathbf{I}$, for any rows $\mathbf{b}_i$, $\mathbf{b}_j$, $\left\|\mathbf{b}_i - \mathbf{b}_j\right\|^2 \leq 2$. Thus, the noise term in the convergence rate of Theorem 4.7 is upper bounded by

$$\frac{\sigma^2}{TL\tau}\left[\frac{1}{\tau}\sum_{t=1}^{T}2 + \sum_{\substack{1\leq t\leq T\\ t=0 \bmod \tau}}2 + 1\right] = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{L\tau^2}\right) = \tilde{\mathcal{O}}\left(L\gamma^2\sigma^2\right)$$

where we used $\tau = \tilde{\mathcal{O}}(1/L\gamma)$. This recovers the tight convergence rate in Proposition 4.5.

## H Experiments

In this section we provide the complete experimental details for the experiments in Section 6, as well as additional experiments on the Stack Overflow dataset.

### H.1 Experiments with Quadratic Functions

We study *random quadratic* function $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{A}\mathbf{x} - \mathbf{b}\right\|^2$ to be able to precisely control the smoothness constant $L$ that appears in our theoretical analysis. In particular, we set the spectrum of $\mathbf{A} \in \mathbb{R}^{100\times 100}$ to have the values to be linearly distributed between $\lambda_{\min} = 0$ and $\lambda_{\max} = \sqrt{L}$,

| Dataset | MNIST | CIFAR-10 | StackOverflow |
|---|---|---|---|
| Train Records | 60,000 | 50,000 | 135,818,730 |
| Test Records | 10,000 | 10,000 | 16,586,035 |
| Dimensionality | 784 | 3,072 | 200,000 |
| Classes | 10 | 10 | 10,000 |
| Model | Logistic | CNN | LSTM |
| Privacy Unit | Example | Example | User |
| Parameters | 7,056 | 550,570 | 4,050,748 |
| Learning Setting | Centralized | Centralized | Federated |

Table 1: Summary of datasets and associated problems considered in this empirical evaluation.

and we randomly shift the axis by unitary transformation. We calculate the unitary transformation by the SVD of a random matrix $\mathbf{D}$ with every element $d_{ij} \in \mathcal{N}(0, 1)$. Lets $\mathbf{D} = \mathbf{U}_D \mathbf{\Lambda}_D \mathbf{V}_D$ be the SVD decomposition, and let $\mathbf{\Lambda}_A = \text{diag}(\lambda_{\max}, \ldots, \lambda_{\min})$ is the matrix with the desired spectrum (between $\lambda_{\min} = 0$ and $\lambda_{\max} = \sqrt{L}$). We calculate the matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}_D \mathbf{\Lambda}_A \mathbf{V}_D$. We also randomly sample the shift $\mathbf{b} \in \mathcal{N}(0, \mathbf{I})$, $\mathbf{b} \in \mathbb{R}^{100}$.

We note that such quadratic function $f$ is $L$-smooth and convex. We fix the number of iterations $T$ to $5000$, and the variance of the noise $\sigma$ is equal to 20.

In these experiments we aim to compare DP-MF, and our proposed DP-MF$^+$ methods under varying hyperparameter settings. We fix the smoothness $L = 10$, and we vary the learning rate $\gamma$ over the logarithmic grid between $10^{-4}$ and 1, and we further select the region of learning rates around the optimal $\gamma$. We also tune parameter $\tau$ in DP-MF$^+$ over the grid $\{1, 2, 10, 50, 100, 200, 500, 1000, 5000\}$.

### H.2 Practical DP Training Experiments

**Datasets and tasks.** Table 1 summarizes the datasets and problems used in our empirical evaluation. For the MNIST dataset, light preprocessing is done so the $28 \times 28$ input images are flattened to size $784$ vectors and normalized so entries lie in the range $[0, 1]$. For the CIFAR-10 and Stack Overflow datasets, the experimental setup including data preprocessing follows exactly from Denisov et al. [10] and Choquette-Choo et al. [7].

**Metrics.** For each dataset, mechanism, and privacy parameter, we run the mechanism for multiple trials and report the test set accuracy of the final iterate. We compute the mean and standard error of the reported test set accuracies.

**MNIST, logistic regression.** For MNIST we train a logistic regression model to predict image labels. All mechanisms train for $T = 2048$ iterations and either 1 or 16 epochs, corresponding to batch sizes of 29 and 469 respectively.[4] We vary $\varepsilon$ over $\{0.01, 0.1, \ldots, 100\}$ and fix $\delta = 10^{-6}$. We fix the clipping threshold at $1.0$ and the learning rate at $0.5$. We run each experiment for $5$ trials, and plot the mean test set accuracy along with error bars indicating the standard error of the estimate.

**CIFAR-10, CNN.** For CIFAR-10, we follow the experimental setup from [7] and train a CNN model to predict image labels. Specifically, we train all mechanisms for 20 epochs and $T = 2000$ iterations, which corresponds to a batch size of $500$.[5] We consider $\varepsilon = 1, 2, 4, 8, 16, 32$ and set $\delta = 10^{-6}$. We tune the learning rate non-privately for each method and $\varepsilon$ by running a single trial with a fixed random seed and choosing the one which achieved the lowest training error. For each value of $\varepsilon$, we use the tuned learning rate and run 12 new trials with different random seeds, and record the test set accuracy at the end of training.

**Stack Overflow, LSTM.** We follow the experimental setup of Denisov et al. [10], and train a next-word prediction LSTM model on the Stack Overflow dataset [3]. We train each mechanism

[4] In practice, one often trains small-scale models for many epochs, perhaps even using full-batch gradients, to improve the privacy/utility trade-off (at the cost of increased computation). We are interested in the *relative* performance for a fixed computation budget, so we train for a small number of epochs.

[5] While Choquette-Choo et al. [7] use momentum and learning rate decay, we omit the use of such techniques as they are orthogonal to our theoretical results.

| Noise Multiplier | DP-MF | DP-MF$^+$($\tau = 2048$) |
|---|---|---|
| 0.341 | $24.63 \pm 0.06$ | $24.58 \pm 0.12$ |
| 0.682 | $23.76 \pm 0.14$ | $23.73 \pm 0.16$ |
| 1.364 | $22.54 \pm 0.11$ | $22.44 \pm 0.08$ |
| 2.728 | $11.51 \pm 12.71$ | $10.42 \pm 13.05$ |
| 5.456 | $0.03 \pm 0.02$ | $0.05 \pm 0.06$ |

Table 2: Comparison of test set accuracies on the Stack Overflow next word prediction task between DP-MF and DP-MF$^+$.

for $1$ epoch and $2048$ iterations, which corresponds to about $167$ clients per round, each holding an average of $\approx 400$ records. We vary the hyper-parameters according to prior work and run $2$ trials for each hyper-parameter setting. We report results for the best hyper-parameters setting of each mechanism. We use federated averaging instead of gradient descent. Additionally, to be consistent with the prior work and to test if our proposed factorizations are compatible with the other types of workloads, we use momentum and learning rate decay. Although the $\mathbf{C}$ matrix was optimized for the Prefix workload, $\mathbf{A} = \mathbf{S}$, it is applied to a variant $\mathbf{A} = \mathbf{S}'$ that incorporates momentum and learning rate decay by setting $\mathbf{B} = \mathbf{S}'\mathbf{C}^{-1}$. More details of how DP-MF and DP-MF$^+$ apply to this setting are available in Denisov et al. [10].

The results are shown in Table 2 for varying the noise multiplier, which corresponds to values of $\varepsilon$ are equal to $\{17.65, 7.6, 3.44, 1.61, 0.76\}$. We see no significant difference between DP-MF and DP-MF$^+$, as the small differences in performance are within the statistical bounds one would expect if they had identical means. At larger noise multipliers, both DP-MF and DP-MF$^+$ exhibit learning instabilities.