Privacy-Conscious Algorithm Design via PAC Privacy

Mayuri Sridhar
MIT CSAIL
Cambridge, MA, 02139
Email: mayuri@mit.edu

Xiaochen Zhu
MIT CSAIL
Cambridge, MA, 02139
Email: xczhu@mit.edu

Srinivas Devadas
MIT CSAIL
Cambridge, MA, 02139
Email: devadas@mit.edu

***Abstract*—As the use of algorithms for daily decision-making has grown, so has the need for *provable privacy guarantees* on their leakage of sensitive data. Classically, privatizing these algorithms is implemented by computing the minimal noise required for a given privacy budget and adding it post-hoc. We instead propose that the algorithm's hyperparameters and its privacy budget can be *jointly optimized* to improve overall privatized utility. In this work, we focus on designing *privacy-conscious algorithms*; that is, designing near-optimal algorithms, in terms of utility, for a given privacy budget under PAC Privacy.**

We integrate the noise required for PAC Privacy guarantees directly into our objective functions for classic algorithms like linear regression and gradient descent. To the best of our knowledge, we are the first to optimize for *privatized performance* under PAC Privacy. We demonstrate how to derive theoretically-optimal parameters to maximize utility under a provided privacy budget. These parameters correspond to regularization strength for linear regression and step sizes for gradient descent optimization of convex and smooth objective functions. We validate our theoretical results experimentally, showing the benefits of privacy-conscious design for ridge regression and regularized logistic regression. Our algorithms maintain near-optimal utility at relaxed privacy budgets and smoothly trade off utility and privacy in stricter settings.

1. Introduction

Privacy in machine learning (ML) has become an increasingly urgent concern. Various ML algorithms are used daily to support decision making across sensitive domains. These models are typically produced from training data that often includes identifying information about individuals. The privacy concern in releasing these models for public use has been extensively studied [1], [2], [3], [4]. A standard line of defense is rule-based ad-hoc data sanitization – e.g., many applications often focus on removing personally identifiable information (PII) from the datasets prior to training [5]. However, this approach is known to be insufficient: PII removal is often prone to re-identification [6] and PII detection itself typically requires complex ML models [7].

This motivates a need for *provable privacy guarantees* in algorithmic design that theoretically limit the information an adversary can infer about any specific data point. In pursuit of this, the standard technique is to use differential privacy (DP) [8], [9]. DP has a long history of providing meaningful guarantees for individuals — broadly speaking, by adding sufficient noise to the output model, we can guarantee that it is impossible to distinguish whether a particular user was part of the training set. Generally, there are two main drawbacks of using DP: computing "sufficient noise" tightly requires significant algorithmic whiteboxing [10] and often, the noise required for privatization adversely affects utility [11], [12].

PAC Privacy [13] addresses the former problem — in particular, PAC Privacy allows for efficient *simulations* to estimate the noise required to privatize arbitrary algorithms. This essentially replaces the manual analysis and whiteboxing required by DP techniques with computational power. Further, recent work [14] suggests that there are cases where PAC privatization does not empirically degrade utility for stable algorithms. That is, [14] demonstrates how to privatize a broad class of algorithms by improving their algorithmic stability, showing that introducing regularization can reduce the noise required for privatization and therefore, achieve better privacy-utility tradeoffs.

A parallel line of research explores the relationship between *generalization*, *robustness*, and *privacy*. Overfitting has been identified as a major source of privacy leakage—models with large generalization gaps are especially vulnerable to membership inference attacks [2]. While robustness has sometimes been viewed as a pathway to privacy, recent studies have shown that robustness alone is not sufficient for protecting individual data [15]. Classic results such as the propose-test-release framework [16] and more recent work on certifiable robustness for private models [17], [18] demonstrate that robustness and privacy share deep structural connections. More recent theory [19], [20] formalizes this relationship, constructing near-optimal black-box transformations between robust and private algorithms.

These efforts follow a common *post-hoc privatization* paradigm: start from stable or robust algorithms and then *add* privacy guarantees by injecting the noise required to privatize them. In contrast, our work advocates for a paradigm shift. Rather than modifying existing algorithms to satisfy privacy constraints, we seek to directly design *privacy-conscious algorithms*. We wish to *quantify and minimize* the utility loss due to privatization.

In particular, we observe that the utility loss from

privatization is not easy to parameterize. In DP, for arbitrary mechanisms $\mathcal{M}$, this is perhaps intrinsic. The worst-case change in distribution may have an *unbounded* impact on utility. However, we show that the PAC Privacy model allows us to *directly integrate our privacy budget into the objective function* for our optimization problem. By doing so, we show how privacy can be treated as part of the bias-variance tradeoff, enabling algorithms that achieve near-optimal utility under a fixed privacy budget.

Contributions. Our key contributions are as follows:

1) We provide the first framework to integrate PAC Privacy constraints into the objective function of optimization problems. In particular, we show how PAC Privacy guarantees can be viewed as a simple generalization of the bias-variance tradeoff in machine learning. This allows us to *directly optimize for privatized utility.*
2) We derive theoretically-optimal regularization parameters for a given privacy budget in the setting of regularized linear regression.
3) We provide the first design and implementation of PAC-private gradient descent on smooth and convex functions. We prove that PAC-private gradient descent converges to a near-optimal solution and provide an algorithm to optimize the step size for a given privacy budget.
4) We provide extensive experimental results on regularized linear regression and logistic regression, demonstrating how *privacy-conscious* algorithm design can improve utility across privacy budgets and datasets.

Paper Organization. The rest of this paper is organized as follows. Section 2 defines PAC Privacy and its security guarantees. Section 3 introduces the concept of *privacy-conscious* algorithm design under PAC Privacy and establishes a general framework to optimize algorithmic parameters for *privatized utility*. Section 4 derives the theoretically-optimal regularization parameters for privatized ridge regression. Section 5 proves the convergence of PAC-private gradient descent and derives the theoretically-optimal step sizes. Experimental results for regularized linear regression and logistic regression are presented in Section 6 to validate our theoretical findings. We present a comprehensive literature review in Section 7 and conclude this paper in Section 8.

2. Preliminaries on PAC Privacy

We first provide an overview of the PAC Privacy framework, via the lens of indistinguishability. In particular, we focus on the classic attack of membership inference [1]. That is, consider the problem of identifying whether a particular datapoint x_0 was used to train a model $\mathcal{M}(X)$. Standard Differential Privacy (DP) adds input-independent noise [8] to ensure that the outputs produced on neighboring datasets (e.g., $X \setminus \{x_0\}$ and $X \cup \{x_0\}$) are statistically indistinguishable. However, DP typically requires *white-box sensitivity analysis* [21], [22], [23] to identify the worst-case perturbation. PAC Privacy instead quantifies indistinguishability with respect to the *distribution of input data* rather than the input-independent worst-case.

Given a data pool $\mathcal{U}$, PAC Privacy constructs a distribution $\mathcal{D}$ over subsets of $\mathcal{U}$. The mechanism output $Y = \mathcal{M}(X)$ is then treated as a random variable where $X \sim \mathcal{D}$. By analyzing $\mathrm{Var}[\mathcal{M}(X)]$ over $\mathcal{D}$, the framework determines the minimum noise required to bound the adversary's success rate for arbitrary inference tasks. We now formalize the PAC Privacy definitions below:

Definition 1 ($(\delta, \rho, \mathcal{D})$ PAC Privacy [13])**.** *Let $\mathcal{M} : \mathcal{X}^* \to \mathcal{O}$ be a function, $\mathcal{D}$ a data distribution, and $\rho(\cdot, \cdot)$ an inference criterion. We say $\mathcal{M}$ satisfies $(\delta, \rho, \mathcal{D})$-PAC Privacy if no adversary, given $\mathcal{M}(X)$ with $X \sim \mathcal{D}$, can produce an estimate $\hat{X}$ such that $\rho(\hat{X}, X) = 1$ with probability at least $1 - \delta$. Here, $1 - \delta$ is the posterior success rate.*

Equivalently, $\mathcal{M}$ is $(\Delta_f \delta, \rho, \mathcal{D})$ PAC-advantage private if the posterior advantage in f-divergence satisfies

$$\Delta_f \delta = \mathcal{D}_f(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) = \delta_o^\rho f\left(\frac{\delta}{\delta_o^\rho}\right) + (1 - \delta_o^\rho) f\left(\frac{1-\delta}{1-\delta_o^\rho}\right),$$

where $(1 - \delta_o^\rho)$ is the optimal prior success rate:

$$\delta_o^\rho = \inf_{X' \in \mathcal{X}^*} \Pr_{X \sim \mathcal{D}}(\rho(X', X) \neq 1).$$

When instantiating $\mathcal{D}_f$ as $\mathcal{D}_{\mathsf{KL}}$, Theorem 1 of [13] bounds the posterior advantage $\Delta_{\mathsf{KL}}\delta$ by the mutual information (MI) between the input X and the output $\mathcal{M}(X)$:

$$\Delta_{\mathsf{KL}}\delta = \mathcal{D}_{\mathsf{KL}}(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) \leq \mathsf{MI}\big(X; \mathcal{M}(X)\big). \tag{1}$$

We can then apply a simplified version of Theorem 1 of [14] to determine the noise required to guarantee a bound on MI.

Theorem 1 (Noise Determination [14])**.** *Given an arbitrary deterministic mechanism $\mathcal{M} : \mathcal{X}^* \to \mathbb{R}$, $X \sim \mathcal{D}$, and $B \in \mathbb{R}_{\geq 0}$. Let $\mathcal{M}_B(X) := \mathcal{M}(X) + \Delta$, where*

$$\Delta \sim \mathcal{N}\Big(0, \frac{1}{2B}\mathsf{Var}[\mathcal{M}(X)]\Big).$$

Then, the output $\mathcal{M}_B(X)$ satisfies $\mathsf{MI}(X; \mathcal{M}_B(X)) \leq B$.

Semantically, PAC Privacy allows us to bound the success rate of *arbitrary adversarial attacks* on the input X after observing the noisy output of a given mechanism $\mathcal{M}$. That is, consider a setting with a secret dataset X and an adversary who aims to infer some information about this dataset. PAC Privacy aims to ensure that the *posterior* advantage of the adversary remains tightly bounded after observing the noisy output $\mathcal{M}_B(X)$.

In particular, consider the setting where the adversary aims to determine whether an individual datapoint x_0 was included; this is the standard Membership Inference Attack (MIA) [1]. For a given prior, in the PAC Privacy model, a strong adversary who knows $\mathcal{D}$ *still* has a bounded posterior success rate (determined by the mutual information budget B) in identifying which subset $X \sim \mathcal{D}$ was used after observing the noisy release $\mathcal{M}_B(X)$. Thus, for appropriate choices of $\mathcal{D}$, we can tightly bound the posterior success rate on identifying whether x_0 was used. We formalize this below.

Definition 2 (Membership Inference Attack)**.** *Given a finite data pool $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ and some processing*

mechanism $\mathcal{M}$, $X \sim \mathcal{D}$ is a subset of $\mathcal{U}$ randomly selected. An informed adversary is asked to return a subset $\hat{X}$ as the membership estimation of X after observing $\mathcal{M}(X)$. We say $\mathcal{M}$ is resistant to $(1-\delta_i)$ individual membership inference for the i-th datapoint u_i, if for an arbitrary adversary, $\Pr_{X\sim\mathcal{D},\hat{X}\leftarrow\mathcal{M}(X)}(\mathbf{1}_{u_i\in X} = \mathbf{1}_{u_i\in\hat{X}}) \leq 1-\delta_i$. Here, $\mathbf{1}_{u_i\in X}$ is the indicator variable for $u_i \in X$, and similarly for $\mathbf{1}_{u_i\in\hat{X}}$.

When $\mathcal{D}$ is a Poisson sampling of $\mathcal{U}$ with $p = 0.5$, the prior on individual membership is $(1-\delta_o^\rho) = 0.5$. $\mathsf{MI}(X;\mathcal{M}_B(X)) \leq B$ allows us to bound the maximal posterior success rate of any MIA attack. In particular,

$$\mathcal{D}_{\mathsf{KL}}(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) = \delta \ln\Big(\frac{\delta}{\delta_o^\rho}\Big) + (1-\delta)\ln\Big(\frac{1-\delta}{1-\delta_o^\rho}\Big).$$

Plugging in $(1-\delta_o^\rho) = 0.5$ and $\mathcal{D}_{\mathsf{KL}}(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) \leq B$ allows us to solve this for the maximal values of $(1-\delta)$. In this work, we investigate B, a privacy budget measured in MI from 2^{-10} to 2^{-2}; this corresponds to posterior success rates on MIA of $\approx 53\%$ to 84%.

Matching the posterior success rates allows meaningful comparisons between DP and PAC Privacy. In particular, ϵ-DP can be converted into a posterior success rate guarantee via the following equation [14], [24]:

$$1 - \delta \leq 1 - 1/(1+e^\epsilon). \tag{2}$$

Thus, a PAC Privacy budget $\mathsf{MI} = 2^{-8}$ corresponds to a posterior success rate of 54% under a prior of 0.5. This posterior success rate, in turn, corresponds $\varepsilon \approx 0.18$ in DP under equivalent MIA guarantees [14] via Equation (2).

While DP and PAC Privacy operate on similar problem settings, they provide semantically different privacy guarantees. In particular, DP provides a *worst-case, input-independent* guarantee for any individual datapoint. In contrast, PAC Privacy guarantees are *instance-specific*, and depend on the input distribution $\mathcal{D}$. This relaxation allows us to tightly bound the noise as a function of $\mathcal{D}$ via simulating variance, rather than input-independent sensitivity.

We observe that large MI values do not provide meaningful guarantees for *individual membership*; however, they may still provide strong guarantees for harder tasks like reconstruction, which may have lower priors. In particular, we observe that a prior of 1% and $B = 1$ provides a posterior success rate of $\approx 36\%$. In Section 6.3, we provide a quantitative comparison of privatized linear regression with PAC Privacy MI and the corresponding DP ϵ.

Composition. So far, we focus on single-dimensional outputs. For multidimensional outputs, Theorem 4.1 of [25] shows that the leakage accumulates linearly:

$$\mathsf{MI}(X_1,\ldots,X_d;\{\mathcal{M}_B(X_j)\}_{j=1}^d) \leq dB.$$

This requires no algorithmic changes outside re-sampling X_j per output dimension. We use this guarantee throughout to both bound the leakage across varying dimensions *and* to bound the leakage across multiple releases. We uniformly allocate the total budget across dimensions/releases and add independent noise accordingly; we leave more sophisticated budget allocation strategies to future work.

When invoking this composition theorem, we note that while the MI accumulates linearly, the prior does not always remain 0.5 for multidimensional outputs. In particular, the MI bound applies to the mutual information across the *set* of X_i's used which are drawn *independently*. Thus, if we want to bound the posterior for X_1 (WLOG, any specific X_i), a Poisson sampling of $\mathcal{D}$ induces a prior of 0.5 for individual MIA for a datapoint $x_0 \in \mathcal{U}$. However, in order to test the membership of x_0 across *any* X_i, the prior becomes $(1-1/2^d)$, as each X_i is an *independent* Poisson sampling. Throughout the body of this work, we consider the adversarial task for a specific X_i, as it was state-of-the-art for black-box privatization via PAC Privacy. However, we note that this is a *weaker* security model than classically considered in ML, where the adversarial task is to identify whether datapoints were *ever* used in the training of a model. We demonstrate how recent work [26] allows us to extend to this harder adversarial setting in Appendix I, with minimal changes to experimental results.

3. Designing for Privacy

3.1. Privacy and Utility

Throughout this work, we focus on the bias-variance tradeoff in machine learning. The seminal work of [27] provides a breakdown of mean-squared error (MSE) in terms of bias and variance. That is, consider a training dataset $(X,Y) \sim \mathcal{D}$. We can then measure the performance of a model $\hat{f}$ trained on (X,Y) via its expected MSE on (x,y) where $y = f^*(x) + \epsilon$ for some underlying model f^* and independent zero-mean error ϵ.

$$\begin{aligned}
&\mathsf{MSE}(\hat{f};x) = \mathbb{E}_{\mathcal{D}}[(\hat{f}(x)-y)^2] \\
&= \mathbb{E}_{\mathcal{D}}[(\hat{f}(x)-f^*(x)-\epsilon)^2] \\
&= (\mathbb{E}_{\mathcal{D}}[\hat{f}(x)] - f^*(x))^2 + \mathsf{Var}_{\mathcal{D}}[\hat{f}(x)] + \mathbb{E}[\epsilon^2] \\
&= \mathsf{Bias}(\hat{f})^2 + \mathsf{Var}(\hat{f}) + \mathsf{error},
\end{aligned} \tag{3}$$

where error corresponds to the noise in the underlying distribution of y that cannot be explained by f^*. Here, we provide a definition of *point-wise* MSE, however, we typically measure MSE over *the complete test dataset*. MSE can also be broken down as:

$$\mathsf{MSE} = \mathsf{MSE}^{\mathsf{param}} + \mathsf{error},$$

where $\mathsf{MSE}^{\mathsf{param}} = \mathsf{Bias}^2 + \mathsf{Var}$. Generally, we can only optimize for $\mathsf{MSE}^{\mathsf{param}}$ since error is a function of the *data distribution*, rather than the model.

Consider an algorithm that outputs an estimator $\hat{f}$ with bias Bias and variance Var, such that $\mathsf{MSE}^{\mathsf{param}}$ is minimized. Typically, bias and variance are inversely related. In general, overfitting corresponds to choosing $\hat{f}$ with low bias and high variance. That is, models which overfit are sensitive to the training data and known to have generalization issues. In contrast, models which underfit are known to have high bias and low variance. That is, $\hat{f}$ may not achieve small error

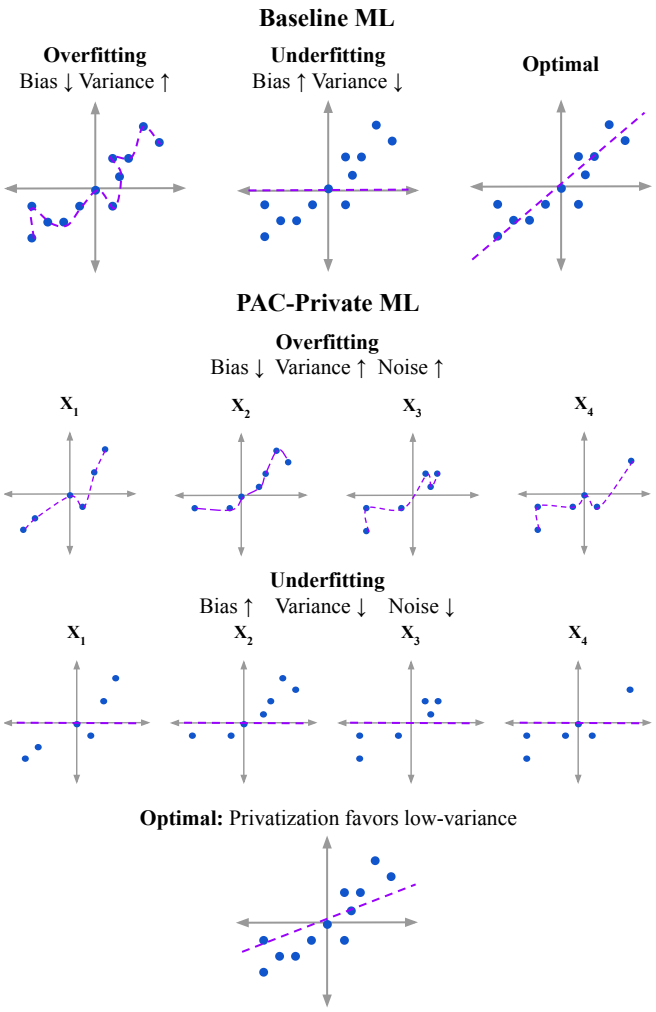

Figure 1. In the non-private setting, we can derive an optimal classifier by trading off bias and variance. For simple models, we can do this by directly optimizing $\mathsf{MSE}^{\mathsf{param}}$. When we integrate PAC Privacy with $X_i \sim \mathcal{D}$, the required noise for privatization depends *only* on the variance of the underlying model. Thus, overfitting is doubly penalized — not only is the generalization error high, the noise for privatization is also large. Designing privacy-conscious algorithms thus favors the low-variance regime.

even on the training set. However, typically, these models generalize better.

Classic techniques like regularization demonstrate how to tradeoff bias and variance to optimize MSE. Theorem 1 implies that privatizing an algorithm increases the *variance* of its output, as determined by the privacy budget. This suggests that the overall MSE for *the privatized algorithm* can be lowered by favoring an output with lower variance at the cost of introducing some bias, as shown in Figure 1.

3.2. PAC Privatization of Arbitrary Estimators

We now consider the utility impact of PAC Privacy. In particular, we begin with an arbitrary algorithm that outputs an estimator $\hat{f}$. We use this estimator to construct a privatized estimator with privacy budget B. Following the ideas of [13], we first need to construct a distribution $\mathcal{D}$, representing the input distribution we sample from. For classic applications in cryptography, this could be the uniform distribution over all n-bit vectors. For machine learning, we consider the universe $\mathcal{U}$ as the complete training dataset. We then construct a random subset $S \subset \mathcal{U}$ via Poisson sampling with $p = 0.5$; $\mathcal{D}$ is then the uniform distribution over all such random subsets.

We measure our *non-private baseline* algorithmic performance via the MSE of the subsampled estimator $\hat{f}_S$. This is not necessarily optimal — that is, the non-private algorithm could use the full dataset, without any subsampling. In practice, on sufficiently large datasets ($|\mathcal{U}|$ large), we observe that $\hat{f}_S$ on random half-subsets performs comparably to the baseline $\hat{f}$ trained on $\mathcal{U}$. This is corroborated by past work in PAC Privacy [14], where the subsampling rarely impacted utility even on small datasets. Denote $\mathsf{MSE}(\hat{f}_S)$ as the MSE of the subsampled estimator in the non-private setting. We can express this as:

$$\mathsf{MSE}(\hat{f}_S) = \mathsf{Bias}(\hat{f}_S)^2 + \mathsf{Var}(\hat{f}_S) + \mathsf{error}.$$

We denote $\hat{f}_{S,B}$ as the estimator with privatization via PAC Privacy with a budget of B; let $\mathsf{MSE}(\hat{f}_{S,B})$ be the mean-squared error of this estimator. To privatize a fixed non-private estimator $\hat{f}_S$ via PAC Privacy with $d = 1$, we analyze the variance of $\hat{f}_S$, and apply Theorem 1. That is, the privatized release is constructed by choosing a noisy estimator where

$$\hat{f}_{S,B} = \hat{f}_S + \Delta \text{ and } \Delta \sim \mathcal{N}\Big(0, \frac{1}{2B}\,\mathsf{Var}[\hat{f}_S]\Big),$$

for $S \sim \mathcal{D}$. We now show how we can express $\mathsf{MSE}(\hat{f}_{S,B})$ in terms of Bias and Var.

Theorem 2. $\mathsf{MSE}(\hat{f}_{S,B})$ *can be decomposed into*

$$\mathsf{MSE}(\hat{f}_{S,B}) = \mathsf{Bias}(\hat{f}_S)^2 + \Big(\frac{1}{2B} + 1\Big)\mathsf{Var}(\hat{f}_S) + \mathsf{error}.$$

Proof. We first consider the change in the estimator due to adding the required noise for privacy.

$$\mathbb{E}[\hat{f}_{S,B}] = \mathbb{E}[\hat{f}_S + \Delta] = \mathbb{E}[\hat{f}_S],$$

since, by definition, the noise Δ is zero-mean. Hence, the overall bias remains unchanged.

We can then compute $\mathsf{Var}[\hat{f}_{S,B}]$:

$$\mathsf{Var}[\hat{f}_{S,B}] = \mathsf{Var}[\hat{f}_S] + \frac{1}{2B}\mathsf{Var}[\hat{f}_S] = \Big(\frac{1}{2B} + 1\Big)\mathsf{Var}[\hat{f}_S],$$

where we note that the noise is generated *independently* of the choice of subset S. Finally, we observe that error is not a function of $\hat{f}$ (rather it only depends on the *true* distribution of y) and thus remains unchanged. □

This provides us with our first insights into designing private algorithms. Broadly speaking, we observe that the error in machine learning can be broken down into bias and variance. In order to optimize for privatized accuracy, the equation simply re-weights the importance of these components. In particular, a privacy budget of B corresponds to amplifying the *variance* portion of MSE by $1/(2B)$.

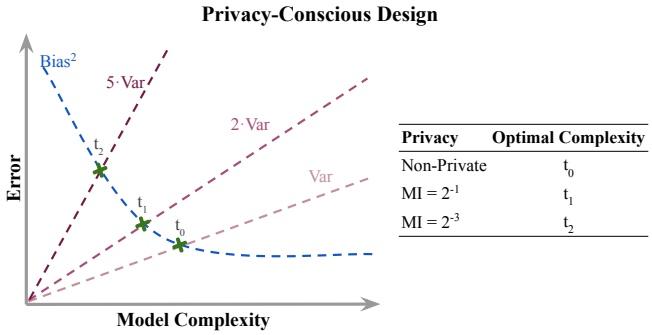

Figure 2. Here, we analyze the optimal MSE at varying levels of privacy. In particular, we observe that there is some irreducible error, due to noise in the system, subsampling etc. We assume that variance grows linearly in model complexity and bias grows inversely. The point t_0 represents the optimal complexity — that is, the complexity that minimizes *non-private* error. However, t_0 does not represent the optimal complexity for privatized algorithms. That is, for a budget B, we must add noise of the form $1/(2B)\mathsf{Var}[\hat{f}_S]$. Specifically, adding noise with variance $\mathsf{Var}[\hat{f}_S]$ provides us with privacy for $\mathsf{MI} = 2^{-1}$; this corresponds to an overall variance of $2\mathsf{Var}[\hat{f}_S]$ for our released estimator. This changes our tradeoff and the optimal point is now at t_1, a lower complexity model. We observe a more extreme version of this phenomenon at stricter privacy budgets, such as $\mathsf{MI} = 2^{-3}$.

3.3. Privacy-Conscious Design

We now consider how to optimize an algorithm for a fixed budget B. Naïvely, without any changes to model complexity, Theorem 2 tells us that the MSE for *any* fixed estimator $\hat{f}_S$ increases by an additive factor of $\mathsf{Var}/(2B)$ due to privatization. We now show how we can design for privacy by considering an algorithmic parameter θ, which allows us to trade off bias and variance, to optimize privatized MSE.

A Concrete Example. Consider a simple setting with error $= 0$, $\mathsf{Bias}^2 \propto \theta$, and $\mathsf{Var} \propto 1/\theta$. Denote θ^*_B as the optimal θ for a privacy budget of B. We will now show that θ^*_∞, the minimizer over θ of $\mathsf{MSE}(\hat{f}_S(\theta))$, which is the optimal non-private parameter, is *not* optimal for a small privacy budget B.

Assume the optimal $\mathsf{MSE}(\hat{f}_S)$ has Bias^2 and Var equal (true up to constant factors in this example) — denote $\hat{f}_S(\theta^*_\infty)$ as the optimal estimator for the non-private setting. Then, our error after post-hoc privatization — i.e., without changing θ from θ^*_∞ — would become

$$\mathsf{MSE}\Big(\hat{f}_{S,B}(\theta^*_\infty)\Big) = \Big(2 + \frac{1}{2B}\Big)\mathsf{Var}[\hat{f}_S(\theta^*_\infty)].$$

However, we can improve upon this — consider $\theta > \theta^*_\infty$ where we control the model complexity such that $\mathsf{Bias}^2(\hat{f}_S(\theta)) = 10 \times \mathsf{Var}[\hat{f}_S(\theta)]$. This suggests that

$$\mathsf{MSE}(\hat{f}_S(\theta)) = 11\mathsf{Var}[\hat{f}_S(\theta)]$$

becomes

$$\mathsf{MSE}\Big(\hat{f}_{S,B}(\theta)\Big) = \Big(11 + \frac{1}{2B}\Big)\mathsf{Var}[\hat{f}_S(\theta)]$$

after privatization.

By the optimality of θ^*_∞,

$$11\mathsf{Var}[\hat{f}_S(\theta)] > 2\mathsf{Var}[\hat{f}_S(\theta^*_\infty)].$$

However, the MSE *after privatization* does not necessarily satisfy the same inequality. In particular, we expect

$$\mathsf{Var}[\hat{f}_S(\theta)] < \mathsf{Var}[\hat{f}_S(\theta^*_\infty)],$$

and thus, when B is sufficiently small,

$$\mathsf{MSE}\Big(\hat{f}_{S,B}(\theta^*_\infty)\Big) \approx \frac{1}{2B}\mathsf{Var}[\hat{f}_S(\theta^*_\infty)]$$

$$\text{and } \mathsf{MSE}\Big(\hat{f}_{S,B}(\theta)\Big) \approx \frac{1}{2B}\mathsf{Var}[\hat{f}_S(\theta)].$$

Therefore, when the privacy budget is small,

$$\mathsf{MSE}\Big(\hat{f}_{S,B}(\theta)\Big) < \mathsf{MSE}\Big(\hat{f}_{S,B}(\theta^*_\infty)\Big).$$

This shows that for sufficiently small B, there exist $\theta > \theta^*_\infty$ that improves the privatized utility, as illustrated by Figure 2. We later show how to find θ^*_B concretely in varying settings.

Constructing Private Estimators. We now formalize two design approaches to construct privatized estimators $\hat{f}_{S,B}$, which satisfy our privacy budget B.

Definition 3 (Post-hoc Privatization)**.** *The post-hoc privatization design constructs an estimator as follows. We first construct $\hat{f}_S$ with the optimal parameters for the non-private domain — generically, denote this set of parameters as θ^*_∞. We then add the noise required for PAC Privacy with a budget B; this corresponds to a linear increase in the overall* MSE *and our privatized estimator now becomes $\hat{f}_{S,B}(\theta^*_\infty)$.*

In contrast, privacy-conscious design *optimizes for privatized utility*.

Definition 4 (Privacy-Conscious Design)**.** *The privacy-conscious design constructs an estimator as follows. We first derive the optimal parameters for the budget B — generically, denote this set of parameters as θ^*_B. We then add the noise required for PAC Privacy and our estimator now becomes $\hat{f}_{S,B}(\theta^*_B)$.*

Remark. *We note that privacy-conscious design often relies on* algorithmic white-boxing*. That is, deriving the optimal θ^*_B typically requires knowledge of how $\hat{f}$ is constructed. In contrast, post-hoc privatization can be done in a black-box manner via PAC Privacy, where* any *algorithm can be efficiently privatized. Characterizing the tradeoffs between the human effort required for privacy-conscious design and its utility benefits is an interesting area of future work.*

In a real-world setting, when given the required privacy budget, B, we optimize the hyperparameters of our algorithm to produce the best-performing algorithm satisfying the privacy budget. We can traverse the bias-variance tradeoff by controlling model complexity (e.g., θ corresponding to ℓ_2 regularization strength). We discuss how we can find theoretically-optimal θ^*_B for regularized linear regression and gradient descent in Section 4 and Section 5, respectively.

Remark. *There are also settings where the algorithm can optimize performance without knowing the true value of B. For instance, we can choose B_0 as our best estimate of B (typically, a lower bound) and derive $\theta^*_{B_0}$. When algorithms are known to be stable, the variance can be* much *smaller than the squared bias near the optimal point. That is, for stable algorithms, we can choose B_0 small with* minimal impacts on overall MSE. *This formalizes the discussion from [14] that stable algorithms can have strong regularization with* little *performance degradation.*

4. Ridge Regression

4.1. Problem Setup

Classic ridge regression, or ℓ_2-regularized linear regression, is formulated as finding the best optimizer $\hat{w}$ satisfying

$$\hat{w} := \arg\min_w \|Y - Xw\|^2 + \lambda \|w\|^2, \tag{4}$$

where $Y \in \mathbb{R}^{n\times 1}$, $X \in \mathbb{R}^{n\times d}$ and $\hat{w} \in \mathbb{R}^{d\times 1}$, and λ is a hyperparameter trading off solution complexity and accuracy [28]. In particular, the algorithm $\mathcal{M}$ takes as input X, Y, and λ and outputs a vector $\hat{w}$. In [28], the authors note how the overall error can be minimized by trading off bias and variance. That is, they suggest practical techniques for choosing non-zero regularization parameters to minimize overall error in realistic distributions.

We follow a similar vein; in particular, we would like to optimize over λ to design for privacy. That is, following Figure 2, we would like to compute the optimal model complexity in terms of λ, as a function of our privacy budget, represented as B. Intuitively, as our privacy budget decreases, the minimal MSE will be achieved at simpler models, corresponding to much larger regularization parameters.

We formalize this intuition via PAC Privacy. To enable privatization, we first construct a distribution $\mathcal{D}$. We follow [13] and consider subsets (X_i, Y_i), which are sampled using Poisson sampling (with $p = 0.5$) from the full training dataset (X, Y). In order to analyze the overall MSE of the private algorithm, we consider $\hat{w}_S$ as the estimator constructed for a subset $S = (X_i, Y_i)$. We can then use the variance across the subsets and Theorem 2 to optimize MSE. For simplicity, we consider the case where $d = 1$; as noted in Section 2, we use independent randomness to treat each dimension separately.

In the following sections, we show how we optimize the MSE of *privatized* algorithms. In particular, we minimize the privatized MSE by optimizing over λ for simple quadratics; we then demonstrate the theoretical improvements of choosing λ in a privacy-conscious manner.

4.2. MSE Analysis

For our analysis, we focus on the nearly-linear setting where $y_i = x_i w^* + \epsilon_i$. That is, the underlying function is linear in X with some noise. We further assume that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$; that is, error $= \sigma^2$. These are standard assumptions for the use of ridge regression. We first consider the non-private setting where our loss is:

$$f(w) = \sum_i (y_i - wx_i)^2 + \lambda w^2.$$

For any λ, this is minimized at

$$\hat{w}(\lambda) = \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \lambda} = \frac{\sum_i x_i^2}{\sum_i x_i^2 + \lambda} w^* + \frac{\sum_i x_i \epsilon_i}{\sum_i x_i^2 + \lambda}.$$

We then consider our *non-private* baseline with PAC subsampling without noise for privatization. Rather than $\hat{w}$ using the complete set X, Y, we construct $\hat{w}_S(\lambda)$, which uses a random subset S via Poisson sampling. That is,

$$\hat{w}_S(\lambda) = \frac{\sum_{i\in S} x_i^2}{\sum_{i\in S} x_i^2 + \lambda} w^* + \frac{\sum_{i\in S} x_i \epsilon_i}{\sum_{i\in S} x_i^2 + \lambda}.$$

Remark. *As noted in Section 3.2, the subsampling may introduce bias* before *privatization. That is, $\hat{w}_S$ is not always an unbiased estimator of $\hat{w}$ — indeed, it is not unbiased for ridge regression. Empirically, we observe that for large n, this bias is typically negligible in terms of overall utility. Throughout this section, we note that our non-private baseline is represented by $\hat{w}_S$ rather than $\hat{w}$.*

We observe that $\hat{w}_S$ is a random variable over the choice of random subset S. We can compute the MSE between the chosen $\hat{w}_S$ and w^*. In particular, by Equation (3), we have

$$\mathsf{MSE}(\hat{w}_S(\lambda)) = \mathsf{Bias}(\hat{w}_S(\lambda))^2 + \mathsf{Var}(\hat{w}_S(\lambda)) + \sigma^2.$$

We now consider the privatized estimator

$$\hat{w}_{S,B}(\lambda) = \hat{w}_S(\lambda) + \Delta.$$

PAC Privacy guarantees (Theorem 1) provide that

$$\mathsf{MI}(S; \hat{w}_{S,B}(\lambda)) \le B, \text{ for } \Delta \sim \mathcal{N}\Big(0, \frac{1}{2B}\mathsf{Var}[\hat{w}_S]\Big).$$

We can then compute the MSE for $\hat{w}_{S,B}$. Directly applying Theorem 2 suggests that for any *fixed* estimator, the variance portion of our error must increase *linearly* in $1/(2B)$. Note that, in the general setting ($d > 1$), the noise per dimension is increased by a factor of d. In the next section, we show how we can use privacy-conscious design to improve upon the naïve post-hoc design. That is, we can choose λ *as a function of* B to optimally trade off utility and privacy.

4.3. Optimizing Privatized Ridge Regression

To design a near-optimal privatized algorithm, we first compute the total bias and variance of $\hat{w}_S(\lambda)$. Throughout this section, we assume that n is sufficiently large and denote $H_S = \sum_{i\in S} x_i^2$.

We first analyze the bias and variance of $\hat{w}_S$ in the *non-private setting*.

Lemma 1. *The bias of $\hat{w}_S$ is*

$$\mathsf{Bias}(\hat{w}_S(\lambda)) = -w^* \lambda \mathbb{E}_S\Big[\frac{1}{H_S + \lambda}\Big].$$

The proof is provided in full in Appendix A.

Lemma 2. *The variance of $\hat{w}_S$ is*

$$\mathsf{Var}(\hat{w}_S(\lambda)) = \sigma^2 \mathbb{E}_S\Big[\frac{H_S}{(H_S+\lambda)^2}\Big] + w^{*2}\mathsf{Var}_S\Big[\frac{H_S}{H_S+\lambda}\Big].$$

The proof is provided in Appendix B. The variance is separated into two components — proportional to the underlying noise in the distribution and the total signal. The variance *decreases* with λ, allowing us to trade off bias and variance by increasing λ.

Canonically [28], the optimal λ^*_∞ that minimizes $\mathsf{MSE}(\hat{w}_S(\lambda))$ for the non-private setting has the form:

$$\lambda^*_\infty = \frac{\sigma^2}{w^{*2}}.$$

We note that a nonzero λ is optimal for the non-private setting — in particular, to minimize MSE, we add *some* regularization to trade off bias and variance. In particular, this corresponds to adding regularization on the order of the inverse of the *signal-to-noise ratio* (SNR) of the system, defined as

$$\mathrm{SNR} := \frac{w^{*2}}{\sigma^2}.$$

For high SNR settings, λ^*_∞ is small as little regularization is needed to minimize MSE. For our post-hoc privatization design (cf. Definition 3); we now construct $\hat{w}_{S,B}(\lambda^*_\infty)$ by first constructing the non-private estimator $\hat{w}_S(\lambda^*_\infty)$ and adding sufficient noise to satisfy our privacy budget B.

We now show how we can construct the optimal λ^*_B for privacy-conscious design. We use the standard assumption that the design matrix is unbounded — that is, we can perfectly estimate w^* with sufficient samples [29].

Theorem 3. *For sufficiently large n, and x_i satisfying $\lim_{n\to\infty}\sum_{i=1}^n x_i^2 = \infty$, and $C = 1/(2B)$,*

$$\lambda^*_B := \arg\min_\lambda \mathsf{MSE}(\hat{w}_{S,B}(\lambda)) \approx \frac{(C+1)\sigma^2}{w^{*2}}.$$

This proof is provided in Appendix C.

We note that λ^*_B does *not* depend on H_S beyond the guarantee that the design matrix is non-trivial. Instead, we observe that λ^*_B depends only on B (through C) and SNR. This formalizes the argument that *stable algorithms are easy to privatize*. When the SNR is large, the required λ^*_B for privacy remains low and the corresponding MSE is low.

Using Theorem 2 and the above lemmas, we can compute the overall MSE of PAC-private estimators with budget B. From Theorem 2, for generic λ, we know that:

$$\begin{aligned}&\mathsf{MSE}(\hat{w}_{S,B}(\lambda))\\ &= \mathsf{Bias}^2(\hat{w}_{S,B}(\lambda)) + \Big(\frac{1}{2B}+1\Big)\mathsf{Var}(\hat{w}_{S,B}(\lambda)) + \sigma^2.\end{aligned}$$

Using this, we can compare the MSE of the privacy-conscious and post-hoc privatization designs over varying privacy budgets. Let $\mu_S = \mathbb{E}[H_S]$. We observe that μ_S/σ^2 is the Fisher information for w^{*2} and w^{*2} is the effect size [29]. We can thus denote

$$Q = \frac{\mu_S w^{*2}}{\sigma^2}$$

as the scaled SNR of the underlying system. We now compare the parameter MSE, i.e., $\mathsf{MSE}^{\mathsf{param}}$, of the post-hoc privatization ($\lambda = \lambda^*_\infty$) compared to the optimal privacy-conscious ($\lambda = \lambda^*_B$) setting. In particular, as before,

$$\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda)) = \mathsf{MSE}(\hat{w}_{S,B}(\lambda)) - \sigma^2.$$

We analyze the improvement in $\mathsf{MSE}^{\mathsf{param}}$ due to privacy-conscious design.

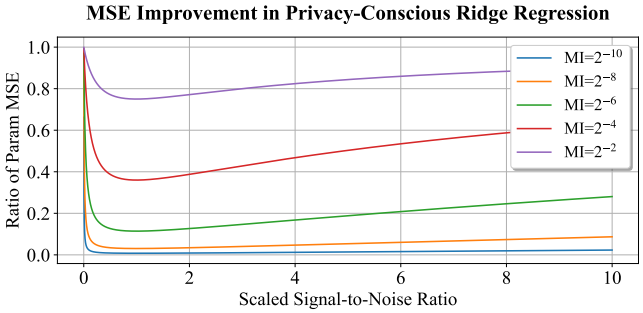

Figure 3. We visualize the results of Corollary 1. In extremely low scaled SNR regimes, linear regression with privacy-conscious design and post-hoc privatization behave similarly — that is, both algorithms have extremely high error and low fidelity. In moderate-high scaled SNR regimes, we can see significant benefits in privacy-conscious design. As our privacy budgets get tighter, the benefits of privacy-conscious design increase.

Corollary 1. *For a privacy budget B, $C = 1/(2B)$, given $\mu_S = \mathbb{E}[H_S]$, the following first-order approximation holds:*

$$\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda^*_\infty)) \approx \frac{\sigma^2 Q(1+(C+1)Q)}{\mu_S(Q+1)^2},$$

$$\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda^*_B)) \approx \frac{Q(C+1)\sigma^2}{\mu_S(Q+C+1)}.$$

The ratio of $\mathsf{MSE}^{\mathsf{param}}$ then becomes:

$$\frac{\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda^*_B))}{\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda^*_\infty))} \approx \frac{(C+1)(Q+1)^2}{(Q+C+1)(QC+Q+1)}.$$

The proof is provided in full in Appendix D. For large n, the first-order approximation typically has negligible error.

We observe that the ratio of $\mathsf{MSE}^{\mathsf{param}}$ is *always* at most 1, since λ^*_B is optimal for the given privacy constraints. Equality is reached only at $C = 0$ (non-private) or $Q = 0$ (no information). We now analyze the effects of privacy-conscious design in varying signal-to-noise regimes.

Our results are summarized in Figure 3. We observe that at extremely low scaled SNR (Q small) regimes, the ratio of MSE is close to 1. This is expected since both the post-hoc privatization design and the privacy-conscious design have very low fidelity. Further, we observe that in these regimes, the primary source of error is due to the irreducible noise (σ^2) in the system. We observe similar behavior for large Q,

where both algorithms have low MSE. This aligns with our theory, where the ratio $\to 1$ as Q tends to either 0 or ∞.

As the SNR increases, we expect the impact of the irreducible noise to become negligible — thus, our overall MSE is well approximated by $\mathsf{MSE}^{\mathsf{param}}$. In this case, we expect to observe significant benefits in $\mathsf{MSE}^{\mathsf{param}}$ due to the privacy-conscious choice of λ_B^*. That is, in simple analytic settings, we observe that even at $\mathsf{MI} = 2^{-4}$, we observe that $\mathsf{MSE}^{\mathsf{param}}$ decreases by at least 40% for $Q \leq 8$ due to privacy-conscious design. This becomes an even stronger effect at tighter MI regimes. For instance, at $\mathsf{MI} = 2^{-10}$, we observe over $40\times$ improvement in $\mathsf{MSE}^{\mathsf{param}}$ even at $Q = 10$. We verify these results on real-world datasets in Section 6.

5. Gradient Descent

In this section, we generalize our results beyond linear regression. In particular, we focus on proving guarantees on PAC-private gradient descent for smooth and convex functions. We show how gradient descent with PAC Privacy converges to a near-optimal solution; we further show how to optimize error via privacy-conscious design.

5.1. Problem Setup

Classic gradient descent (GD) attempts to find the minimum of a smooth function $f : \mathbb{R}^d \to \mathbb{R}$. In machine learning, f is the loss function — in regression, this can correspond to mean-squared error over examples. Functions we consider are of the form:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \phi(w, x_i, y_i). \tag{5}$$

For simple, one-dimensional linear regression, this becomes:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} (wx_i - y_i)^2.$$

Throughout this section, we assume that f is L-smooth and μ-strongly convex *per-example*. This implies that f has a *unique* optimizer w^* where $\nabla f(w^*) = 0$. The L-smoothness guarantee implies that $\forall w_1, w_2$,

$$\|\nabla f(w_1) - \nabla f(w_2)\| \leq L \|w_1 - w_2\|.$$

This provides us with an *upper* bound on the function's gradient. In contrast, μ-convexity states that $\forall w_1, w_2$,

$$f(w_1) \geq f(w_2) + \nabla f(w_1)(w_2 - w_1) + \frac{\mu}{2}\|w_1 - w_2\|_2^2.$$

This provides us with a *lower* bound on the function's gradient. With both assumptions, we can provide theoretical convergence guarantees to gradient descent [30].

In order to enable PAC-private gradient descent, we follow the same strategy as for ridge regression. However, gradient descent involves *many* releases — that is, it is an iterative algorithm. Consider an arbitrary iteration t. At $t = 0$, we choose a fixed initialization at $w_0 = 0$; for $t > 1$, for reasonable stability, we must assume that w_{t-1} is released. This requires us to bound privacy assuming that *each* gradient at time $t > 0$ is released — that is, our privacy budget now scales with dimension *and* horizon.

Consider a fixed horizon T and let $d = 1$. At iteration t, we select a random Poisson-subsampled subset S (i.e., again, each point is included with probability $1/2$). We can then express $\nabla f_S(w_t)$ as the gradient at w_t, evaluated on S:

$$\nabla f_S(w_t) = \frac{1}{|S|}\sum_{(x,y)\in S} \nabla\phi(w_t, x, y).$$

We release the noisy, PAC-private gradient at time t as

$$\hat{g}_{t,B} \;=\; \nabla f_S(w_t) \;+ \Delta_t, \tag{6}$$

where

$$\Delta_t \sim \mathcal{N}\Big(0, \frac{T}{2B}\,\mathsf{Var}_S[\nabla f_S(w_t)]\Big).$$

For simplicity, we denote $B_{\mathsf{epoch}} = B/T$. If $|S| = 0$, we re-sample our subset.

The gradient update is then

$$w_{t+1} \;=\; w_t - \eta \cdot \hat{g}_{t,B}. \tag{7}$$

We can measure the error at time t as $\|e_t\| := \|w_t - w^\star\|$. While we focus on $d = 1$, we can enable general $d > 1$, by applying our algorithm on each dimension independently. As mentioned previously, this increases the required noise per-dimension by a factor of d.

5.2. PAC-Private Gradient Descent Guarantees

We first show that the gradient on a random Poisson-sampled subset remains unbiased. As noted earlier, this is not generally true for arbitrary functions. However, since f is *linear* in the individual datapoints, introducing subsampling does not add bias. We formally prove this below.

Lemma 3. *The PAC-private estimator is unbiased for the sample gradient:*

$$\mathbb{E}[\hat{g}_{t,B} \mid w_t] = \nabla f(w_t).$$

This is proved in Appendix E; intuitively, this is a property of the Poisson estimator on linear functionals. We can then use this lemma to compute the overall error at time t (i.e., $\|e_t\|$) for PAC-private gradient descent. Following standard theory [31], we use step sizes $\leq 1/L$, where L is our smoothness bound on f. Formally, we prove that PAC-private gradient descent retains the linear contraction property of non-private GD. However, there is an additive factor in our error that scales inversely with our privacy budget.

Theorem 4. *At time t, for $\eta \leq 1/L$, for a function f which is μ-strongly convex and L-smooth per-example, with a unique minimizer w^*, PAC-private gradient descent with a per-epoch budget B_{epoch} satisfies*

$$\begin{aligned}\mathbb{E}[\|e_{t+1}\|^2 \mid w_t] \leq\; & (1 - \eta\mu)\,\|e_t\|^2 \\ & + \eta^2(C+1)\mathsf{Var}_S[\nabla f_S(w_t)],\end{aligned} \tag{8}$$

for $C = 1/(2B_{\mathsf{epoch}})$.

Proof. We follow the standard analysis for the error at time t. That is,

$$\begin{aligned}\|e_{t+1}\|^2 &= \|w_{t+1} - w^*\|^2 = \|w_t - \eta\hat{g}_{t,B} - w^*\|^2 \\ &= \|w_t - w^*\|^2 + \|\eta\hat{g}_{t,B}\|^2 - 2\langle w_t - w^*,\, \eta\hat{g}_{t,B}\rangle .\end{aligned}$$

We take the expectation of $\|e_{t+1}\|^2$, conditioned on w_t.

$$\begin{aligned}&\mathbb{E}[\|e_{t+1}\|^2 \mid w_t] \\ &= \mathbb{E}[\|w_t - w^*\|^2] + \mathbb{E}[\|\eta\hat{g}_{t,B}\|^2] - 2\mathbb{E}[\langle w_t - w^*,\, \eta\hat{g}_{t,B}\rangle] \\ &= \|e_t\|^2 + \eta^2\mathbb{E}[\|\hat{g}_{t,B}\|^2] - 2\eta\mathbb{E}[\langle w_t - w^*,\, \hat{g}_{t,B}\rangle].\end{aligned}$$

We suppress the conditioning on w_t for simplicity, however, it is assumed throughout the proof. We first bound $\mathbb{E}[\|\hat{g}_{t,B}\|^2]$. That is,

$$\begin{aligned}&\mathbb{E}[\|\hat{g}_{t,B}\|^2] = \mathbb{E}[\|\nabla f_S(w_t) + \Delta_t\|^2], \\ &= \mathbb{E}_S[\|\nabla f_S(w_t)\|^2] + \mathbb{E}[\|\Delta_t\|^2] + 2\mathbb{E}[\langle\nabla f_S(w_t),\, \Delta_t\rangle] \\ &= \mathbb{E}_S[\|\nabla f_S(w_t)\|^2] + \mathbb{E}[\|\Delta_t\|^2], \\ &= \mathbb{E}_S[\|\nabla f_S(w_t)\|^2] + C\mathsf{Var}_S[\nabla f_S(w_t)]\end{aligned}$$

where we use the fact that Δ_t is centered at 0 and independent of $\nabla f_S(w_t)$. We can then bound

$$\begin{aligned}&\mathbb{E}[\|\nabla f_S(w_t)\|^2] \\ &= \|\nabla f(w_t)\|^2 + \mathsf{Var}_S[\nabla f_S(w_t)] \\ &= \|\nabla f(w_t) - \nabla f(w^*)\|^2 + \mathsf{Var}_S[\nabla f_S(w_t)] \\ &\le L\ \langle w_t - w^*,\, \nabla f(w_t) - \nabla f(w^*)\rangle + \mathsf{Var}_S[\nabla f_S(w_t)] \\ &= L\,\langle w_t - w^*,\, \nabla f(w_t)\rangle + \mathsf{Var}_S[\nabla f_S(w_t)],\end{aligned}$$

where we use the fact that $\nabla f(w^*) = 0$, since w^* is optimal and co-coercivity of gradients for f smooth and convex [32]. Further,

$$\mathbb{E}[\langle w_t - w^*,\, \hat{g}_{t,B}\rangle] = \langle w_t - w^*,\, \nabla f(w_t)\rangle ,$$

via Lemma 3. Plugging these back in to our original equation,

$$\begin{aligned}&\mathbb{E}[\|e_{t+1}\|^2 \mid w_t] \\ &\le \|e_t\|^2 + (\eta^2 L - 2\eta)\,\langle w_t - w^*,\, \nabla f(w_t)\rangle \\ &\quad + \eta^2(C+1)\mathsf{Var}_S[\nabla f_S(w_t)].\end{aligned}$$

If we choose $0 \le \eta \le 1/L$, we have $\eta^2 L - 2\eta \le -\eta$. Further, by μ-strong convexity,

$$\langle w_t - w^*,\, \nabla f(w_t)\rangle \ge \mu\,\|w_t - w^*\|^2 = \mu\,\|e_t\|^2 .$$

Thus,

$$\begin{aligned}\mathbb{E}[\|e_{t+1}\|^2 \mid w_t] &\le (1 - \eta\mu)\,\|e_t\|^2 \\ &\quad + \eta^2(C+1)\mathsf{Var}_S[\nabla f_S(w_t)],\end{aligned}$$

completing our proof. □

Remark. *We observe that the error of PAC-private gradient descent provides similar convergence* rates *to non-private gradient descent. This is expected — we can consider the first term* ($\propto \|e_t\|^2$) *as the* bias *of the error. As we show for linear regression, this does not change for privacy. The variance component* ($\propto \eta^2\mathsf{Var}_S[\nabla f_S(w_t)]$) *is amplified by* $1/(2B_{\mathsf{epoch}})$ *for an overall privacy budget of* TB_{epoch}. *We note that for quadratics, we can get a much stronger guarantee with a factor of* $(1-\eta\mu)^2$, *which resembles our bias-variance tradeoff in Theorem 2 exactly.*

To conclude our general guarantees, we observe that as the number of epochs increases, the error *does not* go to 0. This is generally true for stochastic gradient descent when the variance of batch gradients is non-zero [33]; instead, we converge to a neighborhood of optimality. For a fixed design, at time t, our error increases *linearly* in $1/(2B)$.

5.3. Optimizing Privatized Gradient Descent

To provide tight theoretical guarantees, we focus on a specific class of functions f. That is, in this section, we consider f to be quadratic in w. We are given an overall privacy budget B, measured in MI and a per-epoch budget $B_{\mathsf{epoch}} = B/T$. We first show that, for quadratics, the *final* neighborhood of optimality is asymptotically linear in $1/(2B_{\mathsf{epoch}})$.

Corollary 2. *For quadratic* f, *with* $\sigma^2 = \mathsf{Var}_S[\nabla\hat{f}_S]$, *after* T *iterations, the error from Equation* (8) *becomes:* $\mathbb{E}[\|e_T\|^2 \,|w_{T-1}] \le J_{T,B}(\eta)$, *where*

$$\begin{aligned}J_{T,B} :=&(1-\eta\mu)^T\,\|e_0\|^2 \\ &+ \frac{\eta}{\mu}(C+1)\sigma^2(1-(1-\eta\mu)^T).\end{aligned} \tag{9}$$

As $T \to \infty$,

$$J_{T,B} \to \frac{\eta}{\mu}(C+1)\sigma^2,$$

for any fixed $\eta \le 1/L$ *and* $C = 1/(2B_{\mathsf{epoch}})$.

The proof is provided in Appendix F. We observe that the upper bound on error $\|e_T\|$ is not reducible for *any* constant fixed step size $\eta \le 1/L$. That is, the only way to reduce the error for a fixed horizon T and a given privacy budget B is to reduce η. Thus, in our privacy-conscious design, we focus on optimizing η for a given B and T. Similar to traditional gradient descent, if we have strong guarantees on *the lower bound* of the function (μ-strong convexity), we can make faster progress.

Remark. *We can further optimize allocation of privacy across epochs. This is irrelevant for quadratics where the variance is* fixed *across time. However, this is an important lever when exploring more complex functions where the error may vary with the distance from the optimal point* w^*.

Intuitively, for non-quadratic functions, we must take smaller steps to make progress (in expectation) when the noisy gradient is unstable — typically, we expect this to occur close to the optimal point w^*. *This insight is used in classic learning rate annealing techniques. For instance, the Adam optimizer [34] uses momentum-based techniques to improve the learning rate across epochs. Layering this with*

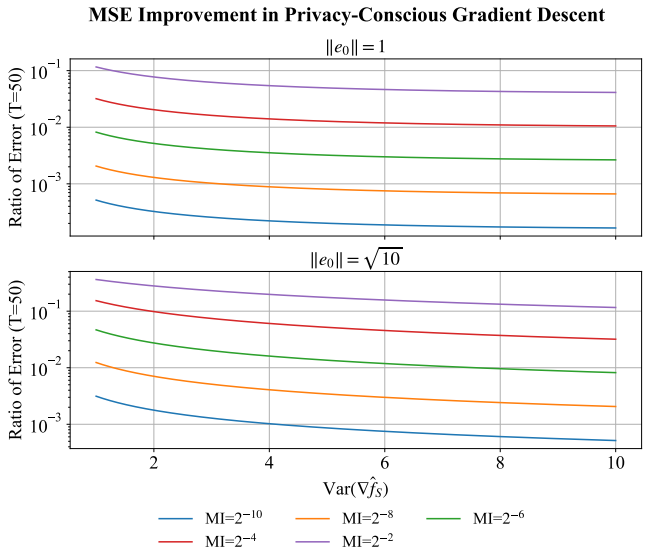

Figure 4. As the underlying variance of ∇f_S (i.e., σ^2 for quadratics) increases, we observe that privacy-conscious gradient descent over $T = 50$ epochs performs *much* better than the post-hoc privatization design. As expected, we observe that the improvement in error increases as our privacy budget tightens. Further, we observe that when our starting bias is *low*, privacy-conscious design has large reductions in error even at low σ^2.

privacy-conscious techniques to improve convergence is an interesting area of future work.

Using Corollary 2, we can derive the optimal η for privacy-conscious design. We consider the error at time T under budget B as a function of η and optimize over η to minimize the finite-horizon error. As suggested earlier, *smaller* step sizes correspond to more stability — intuitively, we can think of this like ℓ_2-regularization [35].

Theorem 5. *For a given privacy budget B, a fixed number of epochs T and $B_{\mathsf{epoch}} = B/T$, the optimal step size minimizing Equation* (9) *for a μ-strongly convex and L-smooth quadratic function f with $\mathsf{Var}_S[\nabla f_S(w_t)] = \sigma^2$, $C = 1/(2B_{\mathsf{epoch}})$, is*

$$\eta_B^* = \arg\min_{\eta} J_{T,B}(\eta)$$
$$\approx \frac{1}{\alpha}\Big[\frac{\alpha\,\|e_0\|^2}{\beta} + 1 - W\Big(\exp\big(\frac{\alpha\,\|e_0\|^2}{\beta} + 1\big)\Big)\Big],$$

where $\alpha = \mu T$, $\beta = (1 + C)\sigma^2/\mu$.

The proof is provided in Appendix G.

As in ridge regression, we compare the privatized utility of η_B^* as defined above to the optimal η_∞^* that minimizes $J_{T,\infty}(\eta)$ in the non-private setting:

$$\eta_\infty^* := \arg\min_{\eta} J_{T,\infty}(\eta)$$
$$\approx \frac{1}{\alpha}\Big[\frac{\alpha\,\|e_0\|^2}{\beta_\infty} + 1 - W\Big(\exp\big(\frac{\alpha\,\|e_0\|^2}{\beta_\infty} + 1\big)\Big)\Big],$$

where $\beta_\infty = \sigma^2/\mu$.

To analyze the benefits of privacy-conscious design, we measure the ratio of the realized upper bounds of $\|e_T\|^2$ between post-hoc privatization (η_∞^*) and privacy-conscious design (η_B^*) with $\mu = L = 1$. For simplicity, we refer to this as the ratio of errors. We consider $T = 50$ and vary the initial bias ($\|e_0\| = 1$ and $\|e_0\| = \sqrt{10}$). We show the ratio of errors over varying σ^2 in Figure 4.

We observe a significant decrease in error due to privacy-conscious design over all regimes. Privacy-conscious algorithms show more performance improvements under tighter MI budgets. Further, we observe stronger improvements (at lower σ^2) when $\|e_0\|$ is small — that is, in regimes where variance is a larger component of the overall error than bias. In all regimes, we observe $\geq 2.7\times$ decrease in error; for $\mathsf{MI} = 2^{-6}$, $\sigma^2 \geq 1$ and $\|e_0\| = 1$, we observe a $100\times$ improvement via privacy-conscious design.

6. Experiments

6.1. Datasets

We give a brief overview of the datasets used throughout the experiments. For linear regression, we use the wine quality dataset [36] and the California Housing dataset [37]. For gradient descent experiments, we use the credit card defaults dataset from Taiwan [38] and MNIST [39].

Wine Quality. The Wine Quality dataset contains 6,497 examples, each of 11 numerical features. The task is to predict wine quality scores, which range from 3 to 9 but are heavily concentrated between 5 and 7, making the dataset moderately imbalanced. The dataset is divided into two subsets — red (1,599 examples) and white (4,898 examples) wine — treated as separate regression tasks due to distinct physicochemical properties and feature distributions [40].

California Housing. The California Housing dataset consists of 20,640 examples, each with 8 numerical features. The task is to estimate the median house value (in units of hundred thousand dollars) in various districts. Features include geographic coordinates, median household size, and median income within each census block group. The target distribution is moderately skewed, with a notable peak at $500,000 due to census processing [37].

Credit Card Defaults. The Credit dataset is a binary classification task. In particular, it consists of 23 features and a single binary label, representing whether the customer will default on their payment next month. While the features are all integers, we note that several features are inherently categorical in nature — for instance, education is stratified into numeric classes representing high school, college, etc. We preprocess these columns using one-hot encoding. The dataset is imbalanced, with 22% of the customers defaulting on their payments.

MNIST. The MNIST dataset is a standard benchmark for handwritten digit classification. It contains 60,000 training and 10,000 test images, each a 28×28 grayscale image with pixel intensity values, flattened into a 784-dimensional

vector. In our experiments, we focus on binary classification variants of MNIST, distinguishing between pairs of digits. We consider two tasks: (1) MNIST (Easy): distinguishing between 0 and 7 and (2) MNIST (Hard): distinguishing between 7 and 9 [41]. Each binary dataset contains roughly 12,000 balanced training examples and 2,000 test examples.

6.2. Experimental Design

For both linear regression and gradient descent experiments, we follow the same experimental design. We first split the dataset into training and test; throughout our experiments, we use 20% of the dataset as the test data unless specified. We note that the MNIST dataset is already split into training and test sets that we do not modify.

We then pre-process our datasets to ensure our independence assumptions. Typically, we first standardize the features to have zero mean and unit variance. We further use standard techniques to orthogonalize the columns [42], [43]; this ensures that the resulting features are not correlated and we can optimize each dimension *independently*. Additionally, for the MNIST dataset, we drop features that contribute less than 5% to the overall variance to improve the stability and computational speed of the underlying algorithm. For categorical features, we use one-hot encodings and drop the first column to remove linear dependence. All transformations are learned on the training data and applied to both the training and test data.

For our linear regression experiments, we then measure the MSE on our test dataset, for MI ranging from 2^{-2} to 2^{-10}. These MI budgets represent *the overall* privacy leakage of the release; as required, the noise per dimension is scaled by d. To privatize a release, we compute the required noise as in Theorem 1; variance is estimated per-dimension with $m = 1024$ subsets, following [14]. Our primary parameter to optimize is the ℓ_2 regularization parameter λ. We compute λ^*_∞ and λ^*_B with and without oracle access to the exact SNR ratio. All MSE results are averaged over 1,000 trials.

For our gradient descent experiments, we train a logistic regression model. To satisfy our theoretical guarantees, we calculate the required L for each dataset. However, we note that the logistic loss function is *not* μ-strongly convex. Thus, we add ℓ_2 regularization $\mu\|w\|^2/2$ to the loss function to ensure μ-strong convexity. We choose μ to ensure that the optimal step size is not affected by the $1/L$ clipping bound. We use PAC-private gradient descent with $T = 50$ to minimize the regularized logistic loss. To compute the noise for privatization, $\mathsf{Var}_S[\nabla f_S(w_t)]$ is evaluated analytically via per-sample gradients. Our primary parameter to optimize is the step size η — we compute η^*_∞ and η^*_B with and without oracle access to the exact $\|e_0\|$ values. Since the regularized loss function is not quadratic, we approximate σ^2 by $\mathsf{Var}_S[\nabla f_S(w_t)]$ to compute the appropriate η via Theorem 5. For classification, we measure the test accuracy for overall MI ranging from 2^{-2} to 2^{-10}, with per-epoch MI scaled by d and T appropriately. All results are averaged over 500 trials.

Throughout the body of this work, we use the composition theorem discussed in Section 2, where our secret subset X_i is re-sampled per dimension and per iteration. As discussed, this is a weaker setting, where the adversarial task for MIA is testing membership in a *particular* iterate, rather than across iterates of the training algorithm. For linear regression, no changes are required to consider the setting where the *same* secret subset is used across dimensions, since it only requires a single release. However, gradient descent requires advanced composition; in Appendix I we demonstrate how to enable this change via recent work [26], while remaining compatible with privacy-conscious design. We observe minimal change in our experimental results due to the change in adversarial models.

All code is available at https://github.com/mayuri95/designing_for_privacy/.

6.3. Ridge Regression Results

We first consider the oracle model where the exact SNR per-dimension is known — this is the best-case scenario for both post-hoc privatization and privacy-conscious design. Our MSE results across the datasets are summarized in Figure 5.

Here, we observe the significant benefits of privacy-conscious design across all datasets. As discussed in Section 4, the benefits of privacy-conscious design increase significantly when the privacy budget is tight — e.g., $\mathsf{MI} \leq 2^{-6}$. That is, as the privacy constraints become stricter, the primary source of error is due to the noise for privatization, rather than the underlying noise of the system. This mirrors our theory, where the optimal choice of regularization favors increasing bias in favor of lowering variance when privacy budgets are tight. While the privacy-conscious design implements this, with negligible error at large privacy budgets, the post-hoc privatization design cannot.

However, we note that knowing the exact SNR may be an unrealistic assumption — thus, we consider a naïve approximation of SNR = 0.1 per dimension. This corresponds to an overall SNR estimate ≈ 1 (1.1 for the Wine datasets and 0.8 for Housing) — a simple regime where signal and noise are considered on the same order of magnitude. Here, we consider 3 designs:

1) λ^*_∞: the post-hoc privatization design.
2) λ^*_B: the privacy-conscious design.
3) λ^*_{DP}: We implement the AdaSSP algorithm [44] which optimizes λ for differential privacy, based on the privacy budget. This *does not* use constant SNR. Instead, AdaSSP uses 3 releases — it first estimates the minimum eigenvalue of X^TX to adaptively estimate λ^*_{DP}. Then, it privately estimates X^TX and X^Ty to estimate the optimal weight vector $\hat{w}$. AdaSSP provides an (ϵ, δ) guarantee with $\delta = 10^{-5}$. In our experiments, we match the theoretical guarantees on MIA posterior success rates provided by DP and PAC Privacy to allow meaningful comparisons [14].

Our results are summarized in Table 1. We observe some increases in error across designs due to the approximation of SNR. However, the optimal privacy-conscious design still

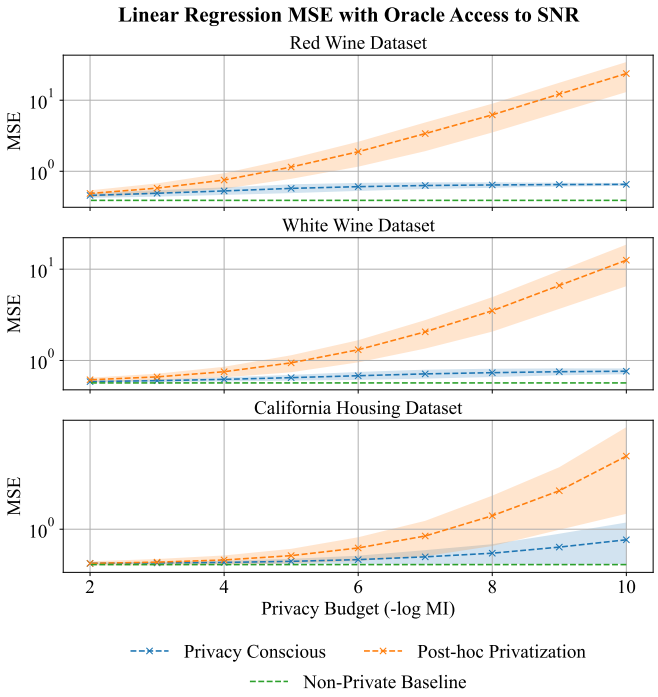

Figure 5. Across all datasets, we see a clear benefit to privacy-conscious design in the oracle model, when the exact per-dimension SNR is used. The overall privacy budget is measured in a negative log scale — that is, $x = 10$ corresponds to an overall budget of $\mathsf{MI} = 2^{-10}$. We measure utility via the mean-squared error between the learned $\hat{w}$ and the test data points. As the overall budget *decreases*, the gap between the privacy-conscious and post-hoc privatization designs widens. When the MI budget is tight, the error in the post-hoc privatization design explodes as it is dominated by the noise required for privatization.

matches or outperforms all other baselines in terms of MSE over all MI. Further, we observe that for larger datasets (e.g., Housing), our overall MSE for λ_B^* nearly matches our non-private baseline at large MI. For smaller datasets, we observe that there is an small initial increase in MSE across both PAC designs — this is partially due to subsampling error. The error from privatization remains low across MI budgets in the privacy-conscious design. We consider other SNR approximations in Appendix H, where we observe that privacy-conscious design matches or outperforms post-hoc privatization across all approximations and datasets.

In general, λ_{DP}^* is comparable to the privacy-conscious design at tight ϵ/MI budgets, but we observe higher loss at large ϵ/MI budgets. We observe that λ_B^* consistently outperforms λ_{DP}^* on the Housing dataset. As we discuss in Section 7, DP and PAC provide semantically different privacy guarantees, so this is not an apples-to-apples comparison.

6.4. Gradient Descent Results

For logistic regression via gradient descent, we first consider an oracle model — that is, the initial bias $\|e_0\|$ is correctly specified per dimension. This represents the best case for both the post-hoc privatization design and the privacy-conscious design. Our results across datasets are summarized in Figure 6.

TABLE 1. WE CONSIDER THE SETTING WHERE THE SNR IS APPROXIMATED AT 0.1 PER DIMENSION. λ_∞^* IS THE POST-HOC PRIVATIZATION DESIGN AND λ_B^* IS THE OPTIMAL PRIVACY-CONSCIOUS DESIGN. WE ALSO CONSIDER THE PRIVACY-CONSCIOUS BASELINE FOR DP [44]. WE REPORT MSE ACROSS MI AND ϵ BUDGETS FOR EACH BASELINE. PRIVACY-CONSCIOUS DESIGN PRODUCES THE BEST RESULTS ACROSS ALL DATASETS AND PRIVACY BUDGETS. SEE TABLE 3 FOR ADDITIONAL RESULTS ON VARYING APPROXIMATIONS OF SNR.

Dataset (Non-Private MSE)		**Privacy Budget**				
	MI **(PAC)**	2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-10}
	ϵ **(DP)**	1.64	0.73	0.36	0.18	0.09
Red Wine (0.39)	λ_∞^*	**0.52**	0.88	2.42	8.26	32.28
	λ_B^*	**0.49**	**0.58**	**0.64**	**0.65**	**0.66**
	λ_{DP}^*	0.60	**0.63**	**0.65**	**0.66**	**0.66**
White Wine (0.57)	λ_∞^*	**0.63**	0.8	1.51	4.32	15.23
	λ_B^*	**0.62**	**0.69**	**0.76**	**0.78**	**0.78**
	λ_{DP}^*	0.76	0.77	**0.77**	**0.77**	**0.78**
Housing (0.56)	λ_∞^*	**0.62**	0.83	1.58	4.84	16.36
	λ_B^*	**0.57**	**0.61**	**0.72**	**0.94**	**1.17**
	λ_{DP}^*	1.26	1.29	1.30	1.31	1.31

On the Credit and MNIST (Easy) datasets, we observe minimal utility loss from the privacy-conscious design for most values of MI. In particular, the Credit dataset achieves $> 90\%$ test accuracy for $\mathsf{MI} \geq 2^{-6}$ and the MNIST (Easy) dataset achieves $> 90\%$ test accuracy for $\mathsf{MI} \geq 2^{-8}$. In these settings, gradient descent can converge after 50 "small" steps, and we thus see minimal losses in test accuracy due to privatization. The MNIST (Hard) dataset is harder to separate [41]. Thus, while we observe the same trend, we see significant utility losses for $\mathsf{MI} \leq 2^{-4}$; here, we observe accuracy $> 76\%$ for $\mathsf{MI} \geq 2^{-6}$. As the MI budgets become tighter, our performance degrades smoothly. For both the Credit and MNIST (Easy) datasets, our test accuracy remains above 70% at all MI. For the MNIST (Hard) dataset, our test accuracy degrades to 57% at $\mathsf{MI} = 2^{-10}$. At all MI values, the privacy-conscious design consistently matches or outperforms the post-hoc privatization design.

As in linear regression, we note that knowing the exact $\|e_0\|$ per dimension is an unrealistic assumption — we thus approximate it with a constant $\|e_0\| = 0.1$ per dimension. To the best of our knowledge, there is no comparable DP baseline that directly optimizes step size for privatized utility (cf. Section 7).

Our results are summarized in Table 2. We observe that privacy-conscious design matches or outperforms post-hoc privatization across all MI and datasets, in terms of test accuracy. As in linear regression, we observe degradation in test accuracy across all designs due to $\|e_0\|$ misspecification compared to the oracle model. This is most clearly seen in the MNIST (Easy) dataset — that is, in this dataset, there is a *single* feature which has high weight where the starting error for this feature is consistently *underestimated* and thus, we

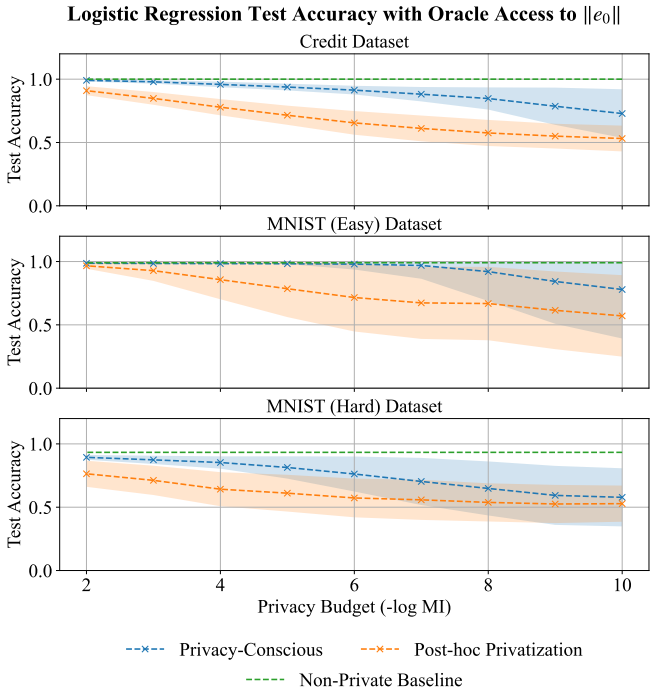

Figure 6. Across all datasets and all MI budgets, we observe that privacy-conscious design consistently outperforms the post-hoc privatization design in the oracle model with known $\|e_0\|$. On relatively easy problems like Credit or MNIST (Easy), privacy-conscious design allows us to privatize algorithms with small impacts on utility for relatively small MI. When the privacy budgets are small, or on more difficult settings like MNIST (Hard), privacy-conscious design smoothly trades off privacy and utility.

observe larger utility drops in the privacy-conscious design compared to the oracle model.

Similar to linear regression, we consider other $\|e_0\|$ approximations in Appendix H. Broadly, our optimal privacy-conscious designs consistently match or outperform the post-hoc privatization design; as our approximations improve, we approach the accuracy under the oracle model. Layering better dimension-specific priors (e.g., via estimators like Adam [34]) to improve privatized performance in practical settings is an interesting area for future work.

7. Related Work

Differential Privacy vs. PAC Privacy. Differential privacy (DP) [8], [45] is arguably the standard framework for privacy-preserving machine learning. DP guarantees that the output of an algorithm changes only marginally when a single individual's data is modified, thereby limiting what can be inferred about any participant. To enable this, DP injects random noise whose magnitude is calibrated to the algorithm's *sensitivity* — the extent to which a single data point can influence the output.

DP and PAC Privacy provide semantically different security guarantees. In particular, DP provides stronger, *input-independent* guarantees compared to PAC Privacy's *instance-specific* ones [13]. However, PAC Privacy bounds the overall divergence across the distribution $\mathcal{D}$ and thus, applies to

TABLE 2. WE CONSIDER THE SETTING WITH A CONSTANT ESTIMATE OF $\|e_0\| = 0.1$. OUR PRIVACY-CONSCIOUS DESIGN CONSISTENTLY OUTPERFORMS POST-HOC PRIVATIZATION. GENERALLY, η_B^* PROVIDES HIGH ACCURACY AT LARGE MI BUDGETS AND SMOOTHLY DEGRADES AS THE PRIVACY BUDGET DECREASES. SEE TABLE 4 FOR ADDITIONAL RESULTS ON VARYING APPROXIMATIONS OF $\|e_0\|$.

Dataset (Non-Private Acc.)	η	**Total Mutual Information Budget**				
		2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-10}
Credit (1.00)	η_∞^*	0.86	0.73	0.62	0.55	0.53
	η_B^*	**0.96**	**0.89**	**0.80**	**0.68**	**0.60**
MNIST (Easy) (0.99)	η_∞^*	**0.95**	0.81	0.67	**0.63**	**0.58**
	η_B^*	**0.99**	**0.97**	**0.83**	**0.67**	**0.58**
MNIST (Hard) (0.93)	η_∞^*	0.76	0.64	**0.56**	**0.54**	**0.52**
	η_B^*	**0.89**	**0.74**	**0.60**	**0.54**	**0.52**

arbitrary adversarial tasks. That is, the only difference in PAC Privacy between protecting an individual and a group is in the *prior adversarial success rate*. In contrast, DP requires composition to extend to groups, which increases ϵ. In this section, we review three lines of work in DP that are most relevant to ours.

Hyperparameter tuning under DP. Hyperparameter tuning under DP is usually performed via evaluating the differentially-private algorithm on a set of candidate hyperparameter configurations, which consumes additional privacy budget. There is a rich line of work on efficiently finding optimal hyperparameter configurations with a diversity of approaches. [46] addresses this issue by tightening privacy accounting via Renyi-DP. [47] uses only a subsample of data to find hyperparameters allowing for privacy amplification. Broadly, [48] provides empirical comparisons of optimization strategies under DP. More recently, [49] introduces a heuristic linear scaling rule that links optimal learning rate and batch size to the privacy budget, enabling efficient privacy-conscious choice of hyperparameters. However, while these techniques are efficient in practice, we note that there is not a direct comparison with analytically-optimal hyperparameters in DP for general learning problems.

Linear regression under DP. Linear regression under DP has been a canonical problem to study the privacy-utility tradeoff. [50] first introduces objective perturbation for regularized empirical risk minimization (ERM) under DP. [51] extends this work to general optimization via the functional mechanism, which approximates the objective via a polynomial representation and then perturbs the coefficients. [52] proposes differentially-private ordinary least squares (OLS) regression, providing confidence intervals for the regression coefficients. More recently, [53] provides sharp minimax bounds for linear regression under DP, showing fundamental limits on the cost of privacy in this setting. Most of these works focus on designing DP algorithms for regression with minimal or fixed regularization.

Our work is most closely related to [44], the baseline we consider in Section 6. This work uses instance-adaptive

mechanisms to optimize regularization and noise in tandem, achieving near-optimal private estimation. However, the approach of [44] relies on complex posterior sampling and confidence intervals and requires constraints on the data distribution for bounded sensitivity. Our formulation achieves a similar goal within the PAC Privacy framework through simpler, closed-form optimization, directly embedding the privacy budget into the regression objective.

Gradient descent under DP. The study of differentially-private GD began with DP-SGD [9], [54], which adds Gaussian noise to clipped gradients at each iteration to satisfy DP. Under convexity and/or smoothness assumptions, subsequent work [55], [56], [57] has established convergence rates for first- and second-order optimization algorithms with DP guarantees. While these methods precisely characterize how DP affects convergence, they treat algorithmic parameters, such as step size and clipping norm, as fixed or heuristically chosen rather than optimized for a given privacy budget.

A parallel line of work has proposed a series of *adaptive* variants of DP-SGD that adjusts algorithmic parameters during training. Earlier papers [58], [59], [60] focus on adaptive clipping strategies to reduce clipping bias in gradient norms. In our work, clipping is not required as the injected noise in PAC Privacy is calibrated to the sampling-induced variance rather than worst-case sensitivity. Subsequent DP work has extended adaptivity to other parameters, including step sizes. In particular, several recent papers aims to develop differentially-private versions of adaptive optimizers such as Adam and AdaGrad [61], [62], [63]. While these methods have interesting theoretical properties, they typically focus on efficiently privatizing adaptive optimization techniques from the non-private setting. In contrast, our work constructs the theoretically-optimal parameters for *privatized utility*.

Variance Reduction Techniques. Optimization under noise has been a central topic in stochastic learning. Classical techniques such as mini-batching [64] improve convergence rates by controlling the stochasticity inherent in gradient estimation. There is a rich line of work on reducing variance in stochastic gradient descent (SGD) — particularly on maintaining the convergence of gradient descent with the efficiency of SGD. Techniques like SAGA [65] and SVRG [66] allow stochastic gradients to converge faster by anchoring to previous estimates (past averaged gradients and full-batch gradients, respectively). [67] improves upon this via a recursive structure, with an efficient inner loop.

Our contribution highlights the importance of variance reduction to privacy-preserving machine learning in the setting of PAC Privacy. In our setting, the injected Gaussian noise – required for PAC Privacy – interacts with the stochastic noise of gradient sampling, altering the effective variance of updates. By jointly optimizing learning rates with noise scales, we show that privacy noise can be treated analogously to optimization noise, allowing principled tradeoffs between convergence, variance, and privacy guarantees. This aligns with recent trends in optimization under noise [68], but extends them to explicitly integrate PAC Privacy constraints.

8. Conclusions

In this work, we show how we can design privacy-conscious algorithms via PAC Privacy. In particular, we show how to integrate the PAC Privacy budget constraints into the objective function of optimization algorithms like regularized linear regression and gradient descent. This allows us to *directly optimize for privatized utility*.

We demonstrate how PAC Privacy constraints can be represented as optimizing for a different point in the bias-variance tradeoff. With this in mind, we derive theoretically-optimal parameters by investigating the appropriate bias-variance tradeoffs for these algorithms. Specifically, for regularized linear regression, we increase the regularization parameter inversely with the required privacy budget and demonstrate how the resulting algorithm minimizes MSE under a given privacy budget. In the same vein, we construct PAC-private gradient descent for convex and smooth functions; we then show how we can reduce the step size to minimize finite-horizon error for a given privacy budget. We validate our results experimentally on real-world datasets, via regularized linear regression and logistic regression for binary classification. There are many interesting directions for future work. To the best of our knowledge, constructing a theoretically-optimal privacy-conscious gradient descent algorithm for DP is an open problem. Another avenue to explore is optimizing gradient descent via other standard relaxations of privacy definitions (e.g., Renyi-DP [69]). Finally, layering privacy-conscious design with other variance reduction techniques for general optimization problems, including other learning algorithms, is an exciting space.

Acknowledgments

We thank the anonymous reviewers and shepherd for their detailed and constructive feedback. The authors are grateful for support from an MIT CSAIL FinTech AI Award.

References

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *S&P*, 2017.

[2] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *CSF*, 2018.

[3] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "SoK: Security and privacy in machine learning," in *EuroS&P*, 2018.

[4] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–34, 2023.

[5] A. Narayanan and V. Shmatikov, "Myths and fallacies of" personally identifiable information"," *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[6] ——, "Robust de-anonymization of large sparse datasets," in *S&P*, 2008.

[7] M. Savkin, T. Ionov, and V. Konovalov, "SPY: Enhancing privacy with synthetic PII detection dataset," in *NAACL Student Research Workshop*, 2025.

[8] C. Dwork, "Differential privacy," in *ICALP*, 2006.

[9] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*, 2016.

[10] X. Xiao and Y. Tao, "Output perturbation with query relaxation," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 857–869, 2008.

[11] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Optimal noise-adding mechanism in additive differential privacy," in *AISTATS*, 2019.

[12] B. Jiang, W. Zhang, D. Lu, J. Du, S. Sharma, and Q. Yan, "Meeting utility constraints in differential privacy: A privacy-boosting approach," in *S&P*, 2025.

[13] H. Xiao and S. Devadas, "PAC Privacy: Automatic privacy measurement and control of data processing," in *CRYPTO*, 2023.

[14] M. Sridhar, H. Xiao, and S. Devadas, "PAC-private algorithms," in *S&P*, 2025.

[15] S. Yeom, I. Giacomelli, A. Menaged, M. Fredrikson, and S. Jha, "Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning," *J. Comput. Secur.*, vol. 28, no. 1, pp. 35–70, 2020.

[16] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *STOC*, 2009.

[17] M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *S&P*, 2019.

[18] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019.

[19] S. B. Hopkins, G. Kamath, M. Majid, and S. Narayanan, "Robustness implies privacy in statistical estimation," in *STOC*, 2023.

[20] H. Asi, J. Ullman, and L. Zakynthinou, "From robustness to privacy and back," in *ICML*, 2023.

[21] J. P. Near and X. He, "Differential privacy for databases," *Found. Trends Databases*, vol. 11, no. 2, pp. 109–225, 2021.

[22] X. Liu, W. Kong, P. Jain, and S. Oh, "DP-PCA: Statistically optimal and differentially private PCA," in *NeurIPS*, 2022.

[23] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin, "Differentially private k-means clustering," in *CODASPY*, 2016.

[24] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *ICML*, 2015.

[25] H. Xiao, "Automated and provable privatization for black-box processing," Ph.D. dissertation, MIT, 2024.

[26] X. Zhu, M. Sridhar, and S. Devadas, "PAC-private responses with adversarial composition," 2026. [Online]. Available: https://arxiv.org/abs/2601.14033

[27] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, 1992.

[28] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[29] G. Casella and R. Berger, *Statistical inference*. Chapman and Hall/CRC, 2024.

[30] G. Garrigos and R. M. Gower, "Handbook of convergence theorems for (stochastic) gradient methods," *arXiv preprint arXiv:2301.11235*, 2023.

[31] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[32] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017.

[33] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, "SGD: General analysis and improved rates," in *ICML*, 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[35] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *ICML*, 2004.

[36] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Wine Quality," UCI Machine Learning Repository, 2009, DOI: https://doi.org/10.24432/C56S3T.

[37] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[38] I.-C. Yeh, "Default of Credit Card Clients," UCI Machine Learning Repository, 2009, DOI: https://doi.org/10.24432/C55S3H.

[39] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[40] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," in *Decision Support Systems*, vol. 47, no. 4, 2009, pp. 547–553.

[41] G. Mayraz and G. E. Hinton, "Recognizing hand-written digits using hierarchical products of experts," in *NIPS*, 2000.

[42] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[43] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1-3, pp. 37–52, 1987.

[44] Y.-X. Wang, "Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain," in *UAI*, 2018.

[45] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[46] N. Papernot and T. Steinke, "Hyperparameter tuning with Renyi differential privacy," in *ICLR*, 2022.

[47] A. Koskela and T. D. Kulkarni, "Practical differentially private hyperparameter tuning with subsampling," in *NeurIPS*, 2023.

[48] A. Priyanshu, R. Naidu, F. Mireshghallah, and M. Malekzadeh, "Poster: Efficient hyperparameter optimization for differentially private deep learning," in *S&P*, 2022.

[49] A. Panda, X. Tang, V. Sehwag, S. Mahloujifar, and P. Mittal, "A new linear scaling rule for differentially private hyperparameter optimization," in *ICML*, 2024.

[50] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *JMLR*, 2011.

[51] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *Proc. VLDB Endow.*, vol. 5, no. 11, pp. 1364–1375, 2012.

[52] O. Sheffet, "Differentially private ordinary least squares," in *ICML*, 2017.

[53] T. T. Cai, Y. Wang, and L. Zhang, "The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy," *Ann. Stat.*, 2021.

[54] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *GlobalSIP*, 2013.

[55] R. Bassily, C. Guzmán, and A. Nandi, "Non-euclidean differentially private stochastic convex optimization," in *COLT*, 2021.

[56] A. Ganesh, M. Haghifam, T. Steinke, and A. Guha Thakurta, "Faster differentially private convex optimization via second-order methods," in *NeurIPS*, 2023.

[57] J. Su, L. Hu, and D. Wang, "Faster rates of differentially private stochastic convex optimization," *JMLR*, 2024.

[58] V. Pichapati, A. T. Suresh, F. X. Yu, S. J. Reddi, and S. Kumar, "AdaCliP: Adaptive clipping for private SGD," *arXiv preprint arXiv:1908.07643*, 2019.

[59] X. Chen, S. Z. Wu, and M. Hong, "Understanding gradient clipping in private SGD: A geometric perspective," in *NeurIPS*, 2020.

[60] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," in *NeurIPS*, 2021.

[61] A. Koskela and A. Honkela, "Learning rate adaptation for differentially private learning," in *AISTATS*, 2020.

[62] H. Asi, J. Duchi, A. Fallah, O. Javidbakht, and K. Talwar, "Private adaptive gradient methods for convex optimization," in *ICML*, 2021.

[63] T. Li, M. Zaheer, K. Z. Liu, S. J. Reddi, H. B. McMahan, and V. Smith, "Differentially private adaptive optimization with delayed preconditioners," in *ICLR*, 2023.

[64] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *NIPS*, 2013.

[65] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *NIPS*, 2014.

[66] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *NIPS*, 2013.

[67] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "SARAH: a novel method for machine learning problems using stochastic recursive gradient," in *ICML*, 2017.

[68] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.

[69] I. Mironov, "Rényi differential privacy," in *CSF*, 2017, pp. 263–275.

[70] M. Sridhar, M. A. Noguera, C. Jain, K. Kristensen, S. Devadas, H. Xiao, and X. Yu, "PAC-private databases," Cryptology ePrint Archive, Paper 2026/238, 2026.

Ethics Considerations

None.

LLM Usage Considerations

LLMs were used for editorial purposes throughout this work and manuscript; all outputs were inspected by the authors to ensure accuracy and originality.

Appendix A. Proof of Lemma 1

Proof. We can derive this directly as a function of the optimal estimator on the subsets.

$$\begin{aligned}
&\mathsf{Bias}(\hat{w}_S) = \mathbb{E}_S[\hat{w}_S] - w^* \\
&= \mathbb{E}_S\Big[\frac{\sum_{i\in S} x_i^2}{\sum_{i\in S} x_i^2 + \lambda} w^*\Big] - w^* + \mathbb{E}_S\Big[\frac{\sum_{i\in S} x_i \epsilon_i}{\sum_{i\in S} x_i^2 + \lambda}\Big] \\
&= \mathbb{E}_S\Big[\frac{\sum_{i\in S} x_i^2}{\sum_{i\in S} x_i^2 + \lambda} w^*\Big] - w^* \\
&= -w^* \lambda \mathbb{E}_S\Big[\frac{1}{\sum_{i\in S} x_i^2 + \lambda}\Big].
\end{aligned}$$

Let $H_S = \sum_{i\in S} x_i^2$ to recover the statement. □

Appendix B. Proof of Lemma 2

Proof. We calculate the total variance using the law of total variance, conditioning on the choice of subset S. That is,

$$\mathsf{Var}[\hat{w}_S] = \mathbb{E}_S[\mathsf{Var}[\hat{w}_S \mid S]] + \mathsf{Var}_S[\mathbb{E}[\hat{w}_S \mid S]].$$

We consider the first term:

$$\begin{aligned}
&\mathbb{E}_S[\mathsf{Var}[\hat{w}_S \mid S]] = \mathbb{E}_S\Big[\mathsf{Var}\Big[\frac{\sum_{i\in S} x_i y_i}{\sum_{i\in S} x_i^2 + \lambda} \mid S\Big]\Big] \\
&= \mathbb{E}_S\Big[\mathsf{Var}\Big[\frac{\sum_{i\in S} x_i^2 w^*}{\sum_{i\in S} x_i^2 + \lambda} \mid S\Big] + \mathsf{Var}\Big[\frac{\sum_{i\in S} x_i \epsilon_i}{\sum_{i\in S} x_i^2 + \lambda} \mid S\Big]\Big] \\
&= \mathbb{E}_S\Big[\frac{1}{(\sum_{i\in S} x_i^2 + \lambda)^2} \mathsf{Var}\Big[\sum_{i\in S} x_i \epsilon_i\Big]\Big] \\
&= \sigma^2 \mathbb{E}_S\Big[\frac{H_S}{(H_S + \lambda)^2}\Big],
\end{aligned}$$

where the third line uses the fact that the variance of $\hat{w}_S$ conditioned on S is *only* due to the noise in the underlying data.

We can similarly analyze the second term:

$$\begin{aligned}
&\mathsf{Var}_S[\mathbb{E}[\hat{w}_S \mid S]] \\
&= \mathsf{Var}_S\Big[\mathbb{E}\Big[\frac{\sum_{i\in S} x_i^2}{\sum_{i\in S} x_i^2 + \lambda} w^* + \frac{\sum_{i\in S} x_i \epsilon_i}{\sum_{i\in S} x_i^2 + \lambda} \mid S\Big]\Big] \\
&= \mathsf{Var}_S\Big[\frac{\sum_{i\in S} x_i^2}{\sum_{i\in S} x_i^2 + \lambda} w^*\Big] \\
&= w^{*2} \mathsf{Var}_S\Big[\frac{H_S}{H_S + \lambda}\Big],
\end{aligned}$$

where we use the fact that $\mathbb{E}[\epsilon] = 0$. □

Appendix C. Proof of Theorem 3

Proof. We return to the breakdown of MSE, in terms of bias and variance.

$$\begin{aligned}
&J_B(\lambda) := \mathsf{MSE}(\hat{w}_{S,B}(\lambda)) \\
&= \mathsf{Bias}(\hat{w}_S)^2 + (C+1)\mathsf{Var}[\hat{w}_S] + \mathsf{error} \\
&= w^{*2}\lambda^2 \Big(\mathbb{E}_S\Big[\frac{1}{H_S + \lambda}\Big]\Big)^2 + (C+1)\sigma^2 \mathbb{E}_S\Big[\frac{H_S}{(H_S+\lambda)^2}\Big] \\
&+ (C+1) w^{*2} \mathsf{Var}_S\Big[\frac{H_S}{H_S + \lambda}\Big] + \mathsf{error} \\
&= w^{*2}\lambda^2 \mathbb{E}_S\Big[\frac{1}{(H_S+\lambda)^2}\Big] - w^{*2}\lambda^2 \mathsf{Var}_S\Big[\frac{1}{H_S+\lambda}\Big] \\
&+ (C+1)\Big(\sigma^2 \mathbb{E}_S\Big[\frac{H_S}{(H_S+\lambda)^2}\Big]\Big) \\
&+ (C+1) w^{*2} \mathsf{Var}_S\Big[\frac{H_S}{H_S + \lambda}\Big] + \mathsf{error} \\
&= \mathbb{E}_S\Big[\frac{w^{*2}\lambda^2 + (C+1)\sigma^2 H_S}{(H_S+\lambda)^2}\Big] \\
&+ w^{*2} C \mathsf{Var}_S\Big[\frac{H_S}{H_S + \lambda}\Big] + \mathsf{error},
\end{aligned}$$

where the last equality uses linearity of expectation and that $\lambda^2 \mathsf{Var}_S\Big[\frac{1}{H_S+\lambda}\Big] = \mathsf{Var}_S\Big[\frac{\lambda}{H_S+\lambda}\Big] = \mathsf{Var}_S\Big[\frac{H_S}{H_S+\lambda}\Big]$.

Note that when n is sufficiently large, $H_S/(H_S+\lambda) \xrightarrow{P} 1$, therefore, we have

$$J_B(\lambda) \approx \mathbb{E}_S\Big[\frac{w^{*2}\lambda^2 + (C+1)\sigma^2 H_S}{(H_S+\lambda)^2}\Big] + \mathsf{error}. \quad (10)$$

We then take the derivative of Eq. 10 w.r.t. λ to minimize the loss.

$$\frac{\partial J_B}{\partial \lambda} =$$
$$\mathbb{E}_S\Big[\frac{2(H_S+\lambda)[(H_S+\lambda)w^{*2}\lambda - (w^{*2}\lambda^2 + (C+1)\sigma^2 H_S)]}{(H_S+\lambda)^4}\Big]$$
$$= 2\mathbb{E}_S\Big[\frac{H_S(w^{*2}\lambda - (C+1)\sigma^2)}{(H_S+\lambda)^3}\Big]$$
$$= 2(w^{*2}\lambda - (C+1)\sigma^2)\mathbb{E}_S\Big[\frac{H_S}{(H_S+\lambda)^3}\Big].$$

We can then set this equal to 0 and solve for λ. This gives us:

$$\lambda_B^* = \frac{(C+1)\sigma^2}{w^{*2}}.$$

□

Appendix D. Proof of Corollary 1

Proof. We can derive this directly from our MSE formula. We first consider λ_∞^*. We use the first-order approximation of Equation (10):

$$\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda_\infty^*)) \approx \frac{w^{*2}\lambda_\infty^{*2} + (C+1)\sigma^2\mu_S}{(\mu_S+\lambda_\infty^*)^2}$$
$$= \frac{w^{*2}\frac{\sigma^4}{w^{*4}} + (C+1)\sigma^2\mu_S}{(\mu_S+\frac{\sigma^2}{w^{*2}})^2}$$
$$= \frac{\sigma^2 Q(1+(C+1)Q)}{\mu_S(Q+1)^2}.$$

We can do the same for λ_B^*.

$$\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda_B^*)) \approx \frac{w^{*2}\lambda_B^{*2} + (C+1)\sigma^2\mu_S}{(\mu_S+\lambda_B^*)^2}$$
$$= \frac{w^{*2}\frac{(C+1)^2\sigma^4}{w^{*4}} + (C+1)\sigma^2\mu_S}{(\mu_S+\frac{(C+1)\sigma^2}{w^{*2}})^2}$$
$$= \frac{\sigma^2 Q(C+1)}{\mu_S(Q+C+1)}.$$

To complete the argument, we can simply take the ratio:

$$\frac{\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda_B^*))}{\mathsf{MSE}^{\mathsf{param}}(\hat{w}_{S,B}(\lambda_\infty^*))} \approx \frac{Q(C+1)\sigma^2}{\mu_S(Q+C+1)}\frac{\mu_S(Q+1)^2}{\sigma^2 Q(1+QC+Q)}$$
$$= \frac{(C+1)(Q+1)^2}{(Q+C+1)(QC+Q+1)}.$$

□

Appendix E. Proof of Lemma 3

Proof. This can be derived directly with $C = 1/(2B_{\mathsf{epoch}})$:

$$\mathbb{E}[\tilde{g}_{t,B}\big|w_t] = \mathbb{E}_S[\nabla f_S(w_t)] + \mathbb{E}[\Delta_t]$$
$$= \mathbb{E}_S[\nabla f_S(w_t)] + 0$$
$$= \sum_{z=1}^{n} \Pr[|S|=z]\mathbb{E}[\nabla f_S(w_t) \,\big|\, |S|=z]$$
$$= \sum_{z=1}^{n} \Pr[|S|=z]\frac{1}{z}\mathbb{E}\Big[\sum_{(x,y)\in S} \nabla\phi(w_t,x,y) \,\big|\, |S|=z\Big]$$
$$= \sum_{z=1}^{n} \Pr[|S|=z]\frac{1}{z}\sum_{(x,y)} \Pr[(x,y)\in S \,\big|\, |S|=z]\mathbb{E}[\nabla\phi(w_t,x,y)]$$
$$= \sum_{z=1}^{n} \Pr[|S|=z]\frac{1}{n}\sum_{(x,y)} \nabla\phi(w_t,x,y)$$
$$= \sum_{z=1}^{n} \Pr[|S|=z]\nabla f(w_t) = \nabla f(w_t).$$

□

Appendix F. Proof of Corollary 2

Proof. We consider this as a simple geometric series, with fixed η. From Theorem 4, we know that

$$\mathbb{E}[\|e_{t+1}\|^2 \mid w_t] \le (1-\eta\mu)\|e_t\|^2 + \eta^2(C+1)\mathsf{Var}_S[\nabla f_S(w_t)].$$

For fixed η and Var_S over t, the second term becomes a geometric series for $T > 0$ where

$$\mathbb{E}[\|e_T\|^2 \,\big|\, w_{T-1}]$$
$$= (1-\eta\mu)^T\|e_0\|^2 + \eta^2(C+1)\sigma^2$$
$$\cdot(1+(1-\eta\mu)+\ldots(1-\eta\mu)^{T-1})$$
$$= (1-\eta\mu)^T\|e_0\|^2 + \eta^2(C+1)\sigma^2\frac{1-(1-\eta\mu)^T}{1-(1-\eta\mu)}$$
$$= (1-\eta\mu)^T\|e_0\|^2 + \frac{\eta}{\mu}(C+1)\sigma^2(1-(1-\eta\mu)^T).$$

We observe that since $\eta\mu < 1$, as $T\to\infty$, $(1-\eta\mu)^T \to 0$. Thus,

$$\lim_{T\to\infty}\mathbb{E}[\|e_T\|^2 \,\big|\, w_{T-1}] = \frac{\eta}{\mu}(C+1)\sigma^2.$$

□

Appendix G. Proof of Theorem 5

Proof. To derive the optimal step size, we minimize:

$$J_{T,B}(\eta) = (1-\eta\mu)^T\|e_0\|^2 + \frac{\eta}{\mu}(C+1)\sigma^2(1-(1-\eta\mu)^T).$$

For our regime, we use the approximation that $(1-\eta\mu)^T \approx \exp(-\eta\mu T)$.

$$J_{T,B}(\eta) = (1-\eta\mu)^T \|e_0\|^2 + \frac{\eta}{\mu}(C+1)\sigma^2(1-(1-\eta\mu)^T)$$
$$\approx \|e_0\|^2 e^{-\eta\mu T} + \frac{\eta(1+C)\sigma^2}{\mu}(1-e^{-\eta\mu T}).$$

We then let $\alpha = \mu T$ and $\beta = \frac{(1+C)\sigma^2}{\mu}$. Using this,

$$\frac{\partial J_{T,B}}{\partial \eta} = -\alpha \|e_0\|^2 e^{-\eta\alpha} + \beta(1-e^{-\eta\alpha}) + \beta\eta\alpha e^{-\eta\alpha}.$$

When we set this to 0, we get

$$\alpha \|e_0\|^2 e^{-\eta\alpha} = \beta(1-e^{-\eta\alpha}) + \beta\eta\alpha e^{-\eta\alpha}$$
$$\iff e^{-\eta\alpha}(\alpha \|e_0\|^2 + \beta - \beta\eta\alpha) = \beta$$
$$\iff e^{-\eta\alpha}(\frac{\alpha \|e_0\|^2}{\beta} + 1 - \eta\alpha) = 1$$
$$\iff \frac{\alpha \|e_0\|^2}{\beta} + 1 = e^{\eta\alpha} + \eta\alpha,$$

where the third line uses the fact that $\beta \neq 0$. This is solved when

$$\eta_B^* = \frac{1}{\alpha}\Big[\frac{\alpha \|e_0\|^2}{\beta} + 1 - W\Big(\exp\big(\frac{\alpha \|e_0\|^2}{\beta} + 1\big)\Big)\Big],$$

where W is the Lambert W function.

□

Appendix H. Additional Experiments

In this section, we first provide additional results across varying SNR estimates for linear regression. Our results are summarized in Table 3. We measure MSE on λ_∞^* (the post-hoc privatization design, as in Definition 3) and on λ_B^* (the privacy-conscious design, as in Definition 4). We observe the same trends as in Section 6 — when SNR is misspecified, the MSE increases across both designs, compared to the oracle model. However, across all SNR estimates, we see consistent improvements in MSE with privacy-conscious design.

We further discuss varying our estimates of $\|e_0\|$ for gradient descent; our results are summarized in Table 4. As in linear regression, we observe significant benefits from privacy-conscious design across all datasets and nearly all $\|e_0\|$ estimates. The only exception is that for MNIST (Hard) — particularly, when $\|e_0\|$ is significantly *underestimated* (i.e., $\|e_0\| = 0.01$) and the privacy budget is *loose* (i.e., MI $\geq 2^{-4}$). In this case, post-hoc privatization slightly outperforms the privacy-conscious design. This is because the privacy-conscious design takes unnecessarily small steps due to underestimating $\|e_0\|$ and does not converge even under loose privacy constraints.

Overall, we see significant benefits from privacy-conscious design across a *wide range* of hyperparameters. This suggests that even when initial parameters are approximated, privacy-conscious design provides a *practical* approach to improving privatized utility.

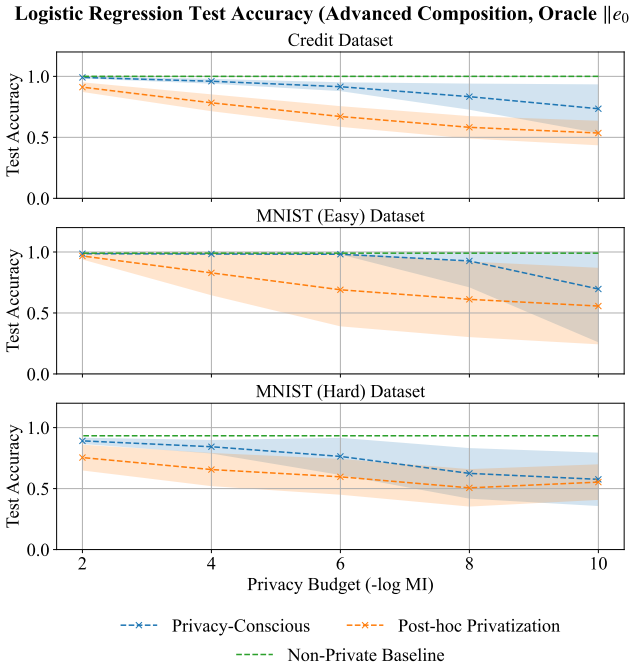

Figure 7. Gradient descent results with advanced composition show minimal changes in accuracy compared to the simple composition setting. Experiments use oracle access to $\|e_0\|$ and accuracy is averaged over 100 trials.

Appendix I. Advanced Composition

As discussed in Section 2, the simple composition theorem (Theorem 4.1 of [25]) focuses on the setting where the choice of the secret subset X_i changes across releases. That is, we bound the mutual information between $X_1 \dots X_d$ and the noisy outputs, in a setting where each X_i is an independent draw from $\mathcal{D}$. While this setting is plausible for certain settings (e.g., database queries for the census [70]), it does not translate into the ML setting. That is, in the ML setting, the standard adversarial task considered is whether individual datapoints are *ever* used to train the output model. As previously discussed, for this adversarial task, under simple composition, the prior grows as $(1-1/2^d)$. At the time of submission, this was state-of-the-art for black-box privatization via PAC Privacy[1].

Recent work in PAC Privacy allows us to consider the setting where the *same* secret is used per iteration via black-box privatization, where the MI remains bounded linearly. Formally, Theorem 3.3 of [26] proves that

$$\mathsf{MI}(X_1, \dots, X_d; \{\mathcal{M}_B(X_j)\}_{j=1}^d) \leq dB,$$

even in the setting where $X_1 = X_2 \cdots = X_d$ when $\mathcal{M}_B$ is calibrated appropriately. That is, while the algorithm still uses instance-specific noise computations, it requires tracking

1. Note that Theorem 4.2 of [25], which allows for the harder adversarial setting is no longer black-box — specifically, this theorem requires a universal upper bound on divergence (similar to DP) making it difficult to integrate with privacy-conscious design.

TABLE 3. LINEAR REGRESSION RESULTS ACROSS VARYING SNR. THE FIRST VALUE OF EACH CELL REPRESENTS THE POST-HOC PRIVATIZATION MSE WITH λ^*_∞ AND THE SECOND VALUE REPRESENTS THE PRIVACY-CONSCIOUS MSE WITH λ^*_B. ACROSS ALL DATASETS AND SNR RATES, WE SEE THE BENEFITS OF PRIVACY-CONSCIOUS DESIGN.

Dataset (Non-Private MSE)	**SNR** (Per-Dim.)	**Total Mutual Information Budget**				
		2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-10}
Red Wine (0.39)	Oracle	(0.49, 0.46)	(0.75, **0.53**)	(1.88, **0.61**)	(6.23, **0.64**)	(23.71, **0.65**)
	10.0	(0.52, 0.52)	(0.90, 0.88)	(2.46, **2.18**)	(8.53, **5.84**)	(31.85, **9.53**)
	1.0	(0.52, 0.51)	(0.90, **0.80**)	(2.33, **1.28**)	(8.48, **1.32**)	(32.30, **0.96**)
	0.1	(0.52, 0.49)	(0.88, **0.58**)	(2.42, **0.64**)	(8.26, **0.65**)	(32.28, **0.66**)
White Wine (0.57)	Oracle	(0.61, 0.59)	(0.75, **0.62**)	(1.31, **0.68**)	(3.51, **0.73**)	(12.54, **0.76**)
	10.0	(0.63, 0.63)	(0.80, 0.81)	(1.52, 1.50)	(4.34, **3.81**)	(15.78, **9.14**)
	1.0	(0.63, 0.63)	(0.81, 0.78)	(1.52, **1.21**)	(4.32, **1.74**)	(15.40, **1.58**)
	0.1	(0.63, 0.62)	(0.80, **0.69**)	(1.51, **0.76**)	(4.32, **0.78**)	(15.23, **0.78**)
Housing (0.56)	Oracle	(0.57, 0.57)	(0.6, 0.58)	(0.73, **0.6**)	(1.25, **0.67**)	(3.35, **0.84**)
	10.0	(0.63, 0.62)	(0.87, 0.82)	(1.65, **1.27**)	(5.50, **2.15**)	(20.87, **4.43**)
	1.0	(0.63, 0.60)	(0.86, **0.67**)	(1.81, **0.84**)	(5.13, **1.36**)	(20.83, **2.25**)
	0.1	(0.62, 0.57)	(0.83, **0.61**)	(1.58, **0.72**)	(4.84, **0.94**)	(16.36, **1.17**)

TABLE 4. GRADIENT DESCENT RESULTS ACROSS VARYING $\|e_0\|$. THE FIRST VALUE OF EACH CELL REPRESENTS THE POST-HOC PRIVATIZATION TEST ACCURACY WITH η^*_∞ AND THE SECOND VALUE REPRESENTS THE PRIVACY-CONSCIOUS TEST ACCURACY WITH η^*_B. WHILE THE TEST ACCURACY IS SENSITIVE TO $\|e_0\|$, WE BROADLY OBSERVE THE BENEFITS OF PRIVACY-CONSCIOUS DESIGN.

Dataset (Non-Private Acc.)	$\|e_0\|$ **(Per-Dim.)**	**Total Mutual Information Budget**				
		2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-10}
Credit (1.00)	Oracle	(0.91, **0.99**)	(0.78, **0.96**)	(0.65, **0.91**)	(0.58, **0.85**)	(0.53, **0.73**)
	0.01	(0.91, **0.98**)	(0.78, **0.92**)	(0.66, **0.81**)	(0.58, **0.69**)	(0.54, **0.61**)
	0.1	(0.86, **0.96**)	(0.73, **0.89**)	(0.62, **0.80**)	(0.55, **0.68**)	(0.53, **0.60**)
	1.0	(0.82, **0.89**)	(0.69, **0.79**)	(0.60, **0.69**)	(0.55, **0.61**)	(0.53, 0.56)
MNIST (Easy) (0.99)	Oracle	(0.97, 0.99)	(0.86, **0.98**)	(0.72, **0.98**)	(0.67, **0.92**)	(0.57, **0.78**)
	0.01	(0.97, 0.96)	(0.90, 0.89)	(0.72, 0.76)	(0.63, 0.64)	(0.55, 0.58)
	0.1	(0.95, 0.99)	(0.81, **0.97**)	(0.67, **0.83**)	(0.63, 0.67)	(0.58, 0.58)
	1.0	(0.92, 0.97)	(0.77, **0.93**)	(0.67, **0.80**)	(0.58, **0.66**)	(0.56, 0.57)
MNIST (Hard) (0.93)	Oracle	(0.76, **0.89**)	(0.64, **0.85**)	(0.57, **0.76**)	(0.54, **0.65**)	(0.53, 0.58)
	0.01	(**0.82**, 0.76)	(**0.70**, 0.64)	(0.59, 0.57)	(0.54, 0.53)	(0.52, 0.52)
	0.1	(0.76, **0.89**)	(0.64, **0.74**)	(0.56, 0.6)	(0.54, 0.54)	(0.52, 0.52)
	1.0	(0.70, **0.83**)	(0.61, **0.73**)	(0.56, **0.63**)	(0.53, 0.56)	(0.51, 0.53)

the posterior belief-state at each iteration; in particular, the noise at iteration i must be calibrated to the adversary's current posterior belief (cf. Algorithm 1 of [26]).

Implementing this algorithm requires minimal changes to our privacy-conscious design framework. Tracking the posterior belief state requires us to estimate variance with a finite number of subsets, rather than analytically. We then bound *conditional* MI at each iteration i by computing variance, weighted by the current adversarial belief state. We use $m = 1024$ subsets and re-implement our gradient descent experiments under the stronger adversarial model.

Our results are summarized in Figure 7. For computational efficiency, we measure accuracy over 5 privacy budgets: $-\log \mathsf{MI} \in [2, 4, 6, 8, 10]$ and run 100 trials per privacy budget. We note that there is some noise in our results due to the lower number of trials. However, the changes in accuracy from the stronger adversarial model are minimal; that is, our results in Figure 7 across datasets are comparable to Figure 6. Again, we consistently observe improvement in accuracy due to privacy-conscious design across all settings. This suggests that privacy-conscious design extends smoothly to the stronger adversarial setting and provides meaningful privacy guarantees in the machine learning context.

Appendix J.
Meta-Review

The following meta-review was prepared by the program committee for the 2026 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

J.1. Summary

The paper builds on top of the concept of PAC privacy which limits the learnable information about a dataset from the model. The paper particularly develops new privacy-preserving learning algorithms, for linear regression and gradient descent, that improve over the state-of-the-art.

J.2. Scientific Contributions

- Creates a New Tool to Enable Future Science
- Addresses a Long-Known Issue
- Provides a Valuable Step Forward in an Established Field
- Establishes a New Research Direction

J.3. Reasons for Acceptance

1) The concept of PAC privacy seems an interesting research direction with the potential to overcome some limitations in privacy-preserving machine learning. It deserves some attention and the proposed algorithm advances this field.

J.4. Noteworthy Concerns

The work builds on the concept of PAC Privacy. It may be helpful to read [13] first.