In 28th *USENIX Security Symposium*, Santa Clara, CA, August 2019

Evaluating Differentially Private Machine Learning in Practice

Bargav Jayaraman and David Evans
Department of Computer Science
University of Virginia

Abstract

Differential privacy is a strong notion for privacy that can be used to prove formal guarantees, in terms of a privacy budget, ϵ, about how much information is leaked by a mechanism. When used in privacy-preserving machine learning, the goal is typically to limit what can be inferred from the model about individual training records. However, the calibration of the privacy budget is not well understood. Implementations of privacy-preserving machine learning often select large values of ϵ in order to get acceptable utility of the model, with little understanding of the impact of such choices on meaningful privacy. Moreover, in scenarios where iterative learning procedures are used, relaxed definitions of differential privacy are often used which appear to reduce the needed privacy budget but present poorly understood trade-offs between privacy and utility. In this paper, we quantify the impact of these choices on privacy in experiments with logistic regression and neural network models. Our main finding is that there is no way to obtain privacy for free—relaxed definitions of differential privacy that reduce the amount of noise needed to improve utility also increase the measured privacy leakage. Current mechanisms for differentially private machine learning rarely offer acceptable utility-privacy trade-offs for complex learning tasks: settings that provide limited accuracy loss provide little effective privacy, and settings that provide strong privacy result in useless models.

1 Introduction

Differential privacy has become a de facto privacy standard, and nearly all works on privacy-preserving machine learning use some form of differential privacy. These works include designs for differentially private versions of prominent machine learning algorithms including empirical risk minimization [11, 12] and deep neural networks [1, 60].

While many methods for achieving differential privacy have been proposed, it is not well understood how to use these methods in practice. In particular, there is little concrete guidance on how to choose an appropriate privacy budget ϵ, and limited understanding of how variants of the differential privacy definition impact privacy in practice. As a result, privacy-preserving machine learning implementations tend to choose arbitrary values for ϵ as needed to achieve acceptable model utility. For instance, the implementation of Shokri and Shmatikov [60] requires ϵ proportional to the size of the target deep learning model, which could be in the order of few millions. Setting ϵ to such arbitrarily large values severely undermines privacy, although there is no consensus on a hard threshold value for ϵ above which formal guarantees differential privacy provides become meaningless in practice.

One proposed way to improve utility for a given privacy budget is to relax the definition of differential privacy. Several relaxed definitions of differential privacy have been proposed that are shown to provide better utility even for small ϵ values [9, 18, 49]. How much privacy leakage these relaxations allow in adversarial scenarios, however, is not well understood. We shed light on this question by evaluating the relaxed differential privacy notions for different choices of ϵ values and empirically measuring privacy leakage, including how many individual training records are exposed by membership inference attacks on different models.

Contributions. Our main contribution is the evaluation of differential privacy mechanisms for machine learning to understand the impact of different choices of ϵ and different relaxations of differential privacy on both utility and privacy. We focus our evaluation on gradient perturbation mechanisms, which are applicable to a wide class of machine learning algorithms including empirical risk minimization (ERM) algorithms such as logistic regression and deep learning (Section 2.2). Our experiments cover three popular differential privacy relaxations: differential privacy with advanced composition, zero-concentrated differential privacy [9], and Rényi differential privacy [49] (described in Section 2.1). These variations allow for tighter analysis of cumulative privacy loss, thereby reducing the noise that must be added in the training process. We evaluate the concrete privacy loss of these variations using

membership inference attacks [61, 74] and attribute inference attacks [74] (Section 3). While the model utility increases with the privacy budget, increasing the privacy budget also increases the success rate of inference attacks. Hence, we aim to find the range of values of ϵ which achieves a balance between utility and privacy, and also to evaluate the concrete privacy leakage in terms of the number of individual members of the training data at risk of exposure. We study both logistic regression and neural network models, on two multi-class classification data sets. Our key findings (Section 4) quantify the practical risks of using different differential privacy notions across a range of privacy budgets.

Related work. Orthogonal to our work, Ding et al. [13] and Hay et al. [26] evaluate the existing differential privacy implementations for the *correctness* of implementation. Whereas, we assume correct implementations and aim to evaluate the impact of the privacy budget and choice of differential privacy variant. While Carlini et al. [10] also explore the effectiveness of differential privacy against attacks, they do not explicitly answer what values of ϵ should be used nor do they evaluate the privacy leakage of the relaxed definitions. Li et al. [42] raise concerns about relaxing the differential privacy notion in order to achieve better overall utility, but do not evaluate the leakage. We perform a thorough evaluation of the differential privacy variations and quantify their leakage for different privacy budgets. The work of Rahman et al. [58] is most closely related to our work. It evaluates differential privacy implementations against membership inference attacks, but does not evaluate the privacy leakage of relaxed variants of differential privacy. Ours is the first work to experimentally measure the excess privacy leakage due to the relaxed notions of differential privacy.

2 Differential Privacy for Machine Learning

Next, we review the definition of differential privacy and its relaxed variants. Section 2.2 surveys mechanisms for achieving differentially private machine learning. Section 2.3 summarizes applications of differential privacy to machine learning and surveys implementations' choices about privacy budgets.

2.1 Background on Differential Privacy

Differential privacy is a probabilistic privacy mechanism that provides an information-theoretic security guarantee. Dwork [16] gives the following definition:

Definition 2.1 ((ϵ, δ)-Differential Privacy)**.** Given two neighboring data sets D and D' differing by one record, a mechanism $\mathcal{M}$ preserves (ϵ, δ)-differential privacy if

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

where ϵ is the privacy budget and δ is the failure probability.

When $\delta = 0$ we achieve a strictly stronger notion of ϵ-differential privacy.

The quantity

$$\ln \frac{Pr[\mathcal{M}(D) \in S]}{Pr[\mathcal{M}(D') \in S]}$$

is called the *privacy loss*.

One way to achieve ϵ-DP and (ϵ, δ)-DP is to add noise sampled from Laplace and Gaussian distributions respectively, where the noise is proportional to the *sensitivity* of the mechanism $\mathcal{M}$:

Definition 2.2 (Sensitivity)**.** For two neighboring data sets D and D' differing by one record, the sensitivity of $\mathcal{M}$ is the maximum change in the output of $\mathcal{M}$ over all possible inputs:

$$\Delta\mathcal{M} = \max_{D, D', \|D - D'\|_1 = 1} \|\mathcal{M}(D) - \mathcal{M}(D')\|$$

where $\|\cdot\|$ is a norm of the vector. Throughout this paper we assume ℓ_2-sensitivity which considers the upper bound on the ℓ_2-norm of $\mathcal{M}(D) - \mathcal{M}(D')$.

Composition. Differential privacy satisfies a simple composition property: when two mechanisms with privacy budgets ϵ_1 and ϵ_2 are performed on the same data, together they consume a privacy budget of $\epsilon_1 + \epsilon_2$. Thus, composing multiple differentially private mechanisms leads to a linear increase in the privacy budget (or corresponding increases in noise to maintain a fixed ϵ total privacy budget).

Relaxed Definitions. Dwork [17] showed that this linear composition bound on ϵ can be reduced at the cost of *slightly* increasing the failure probability δ. In essence, this relaxation considers the linear composition of *expected* privacy loss of mechanisms which can be converted to a cumulative privacy budget ϵ with high probability bound. Dwork defines this as the *advanced composition theorem*, and proves that it applies to any differentially private mechanism.

Three commonly-used subsequent relaxed versions of differential privacy are Concentrated Differential Privacy [18], Zero Concentrated Differential Privacy [9], and Rényi Differential Privacy [49]. All of these achieve tighter analysis of cumulative privacy loss by taking advantage of the fact that the privacy loss random variable is strictly centered around an *expected* privacy loss. The cumulative privacy budget obtained from these analyses bounds the worst case privacy loss of the composition of mechanisms with all but δ failure probability. This reduces the noise required and hence improves utility over multiple compositions. However, it is important to consider the actual impact these relaxations have on the privacy leakage, which is a main focus of this paper.

Dwork et al. [18] note that the privacy loss of a differentially private mechanism follows a sub-Gaussian distribution. In other words, the privacy loss is strictly distributed around the expected privacy loss and the spread is controlled by the variance of the sub-Gaussian distribution. Multiple compositions of differentially private mechanisms thus result in the

	Advanced Comp.	Concentrated (CDP)	Zero-Concentrated (zCDP)	Rényi (RDP)
Expected Loss	$\epsilon(e^\epsilon - 1)$	$\mu = \frac{\epsilon(e^\epsilon-1)}{2}$	$\zeta + \rho = \frac{\epsilon^2}{2}$	$2\epsilon^2$
Variance of Loss	ϵ^2	$\tau^2 = \epsilon^2$	$2\rho = \epsilon^2$	ϵ^2
Convert from ϵ-DP	-	$(\frac{\epsilon(e^\epsilon-1)}{2}, \epsilon)$-CDP	$(\frac{\epsilon^2}{2})$-zCDP	(α, ϵ)-RDP
Convert to DP	-	$(\mu + \tau\sqrt{2\log(1/\delta)}, \delta)$-DP†	$(\zeta + \rho + 2\sqrt{\rho\log(1/\delta)}, \delta)$-DP	$(\epsilon + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP
Composition of k ϵ-DP Mechanisms	$(\epsilon\sqrt{2k\log(1/\delta)} + k\epsilon(e^\epsilon - 1), \delta)$-DP	$(\epsilon\sqrt{2k\log(1/\delta)} + k\epsilon(e^\epsilon - 1)/2, \delta)$-DP	$(\epsilon\sqrt{2k\log(1/\delta)} + k\epsilon^2/2, \delta)$-DP	$(4\epsilon\sqrt{2k\log(1/\delta)}, \delta)$-DP‡

Table 1: Comparison of Different Variations of Differential Privacy
Advanced composition is an implicit property of DP and hence there is no conversion to and from DP.
†. Derived indirectly via zCDP. ‡. When $\log(1/\delta) \geq \epsilon^2 k$.

aggregation of corresponding mean and variance values of the individual sub-Gaussian distributions. This can be converted to a cumulative privacy budget similar to the advanced composition theorem, which in turn reduces the noise that must be added to the individual mechanisms. The authors call this *concentrated differential privacy* [18]:

Definition 2.3 (Concentrated Differential Privacy (CDP))**.** A randomized algorithm $\mathcal{M}$ is (μ, τ)-concentrated differentially private if, for all pairs of adjacent data sets D and D',

$$\mathcal{D}_{subG}(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq (\mu, \tau)$$

where the sub-Gaussian divergence, $\mathcal{D}_{subG}$, is defined such that the expected privacy loss is bounded by μ and after subtracting μ, the resulting centered sub-Gaussian distribution has standard deviation τ. Any ϵ-DP algorithm satisfies $(\epsilon \cdot (e^\epsilon - 1)/2, \epsilon)$-CDP, however the converse is not true.

A variation on CDP, *zero-concentrated differential privacy* (zCDP) [9] uses Rényi divergence as a different method to show that the privacy loss random variable follows a sub-Gaussian distribution.

Definition 2.4 (Zero-Concentrated Differential Privacy (zCDP))**.** A randomized mechanism $\mathcal{M}$ is (ξ, ρ)-zero-concentrated differentially private if, for all neighbouring data sets D and D' and all $\alpha \in (1, \infty)$,

$$\mathcal{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \xi + \rho\alpha$$

where $\mathcal{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D'))$ is the α-Rényi divergence between the distribution of $\mathcal{M}(D)$ and the distribution of $\mathcal{M}(D')$.

$\mathcal{D}_\alpha$ also gives the α-th moment of the privacy loss random variable. For example, $\mathcal{D}_1$ gives the first order moment which is the mean or the expected privacy loss, and $\mathcal{D}_2$ gives the second order moment or the variance of privacy loss. There is a direct relation between DP and zCDP. If $\mathcal{M}$ satisfies ϵ-DP, then it also satisfies $(\frac{1}{2}\epsilon^2)$-zCDP. Furthermore, if $\mathcal{M}$ provides ρ-zCDP, it is $(\rho + 2\sqrt{\rho\log(1/\delta)}, \delta)$-DP for any $\delta > 0$.

The Rényi divergence allows zCDP to be mapped back to DP, which is not the case for CDP. However, Bun and Steinke [9] give a relationship between CDP and zCDP, which allows an indirect mapping from CDP to DP (Table 1).

The use of Rényi divergence as a metric to bound the privacy loss leads to the formulation of a more generic notion of Rényi differential privacy that is applicable to any individual moment of privacy loss random variable:

Definition 2.5 (Rényi Differential Privacy (RDP) [49])**.** A randomized mechanism $\mathcal{M}$ is said to have ϵ-Rényi differential privacy of order α (which can be abbreviated as (α, ϵ)-RDP), if for any adjacent data sets D, D' it holds that

$$\mathcal{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \epsilon.$$

The main difference is that CDP and zCDP linearly bound *all* positive moments of privacy loss, whereas RDP bounds one moment at a time, which allows for a more accurate numerical analysis of privacy loss [49]. If $\mathcal{M}$ is an (α, ϵ)-RDP mechanism, it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$-DP for any $0 < \delta < 1$.

Table 1 compares the relaxed variations of differential privacy. For all the variations, the privacy budget grows sub-linearly with the number of compositions k.

Moments Accountant. Motivated by relaxations of differential privacy, Abadi et al. [1] propose the *moments accountant* (MA) mechanism for bounding the cumulative privacy loss of differentially private algorithms. The moments accountant keeps track of a bound on the moments of the privacy loss random variable during composition. Though the authors do not formalize this as a relaxed definition, their definition of the moments bound is analogous to the Rényi divergence [49]. Thus, the moments accountant can be considered as an instantiation of Rényi differential privacy. The moments accountant is widely used for differentially private deep learning due to its practical implementation in the TensorFlow Privacy library [2] (see Section 2.3 and Table 4).

2.2 Differential Privacy Methods for ML

This section summarizes methods for modifying machine learning algorithms to satisfy differential privacy. First, we

Data: Training data set (X, y)
Result: Model parameters θ
$\theta \leftarrow$ Init(0)
#1. Add noise here: *objective perturbation*
$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, X_i, y_i) + \lambda R(\theta) + \beta$
for *epoch* **in** *epochs* **do**
  #2. Add noise here: *gradient perturbation*
  $\theta = \theta - \eta(\nabla J(\theta) + \beta)$
end
#3. Add noise here: *output perturbation*
return $\theta + \beta$

Algorithm 1: Privacy noise mechanisms.

review convex optimization problems, such as empirical risk minimization (ERM) algorithms, and show several methods for achieving differential privacy during the learning process. Next, we discuss methods that can be applied to non-convex optimization problems, including deep learning.

ERM. Given a training data set (X, y), where X is a feature matrix and y is the vector of class labels, an ERM algorithm aims to reduce the convex objective function of the form,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, X_i, y_i) + \lambda R(\theta),$$

where $\ell(\cdot)$ is a convex loss function (such as mean square error (MSE) or cross-entropy loss) that measures the training loss for a given θ, and $R(\cdot)$ is a regularization function. Commonly used regularization functions include ℓ_1 penalty, which makes the vector θ sparse, and ℓ_2 penalty, which shrinks the values of θ vector.

The goal of the algorithm is to find the optimal θ^* that minimizes the objective function: $\theta^* = \arg\min_\theta J(\theta)$. While many first order [14, 37, 57, 76] and second order [40, 43] methods exist to solve this minimization problem, the most basic procedure is gradient descent where we iteratively calculate the gradient of $J(\theta)$ with respect to θ and update θ with the gradient information. This process is repeated until $J(\theta) \approx 0$ or some other termination condition is met.

There are three obvious candidates for where to add privacy-preserving noise during this training process, demarcated in Algorithm 1. First, we could add noise to the objective function $J(\theta)$, which gives us the *objective perturbation mechanism* (#1 in Algorithm 1). Second, we could add noise to the gradients at each iteration, which gives us the *gradient perturbation mechanism* (#2). Finally, we can add noise to θ^* obtained after the training, which gives us the *output perturbation mechanism* (#3). While there are other methods of achieving differential privacy such as input perturbation [15], sample-aggregate framework [51], exponential mechanism [48] and teacher ensemble framework [52]. We focus our experimental analysis on gradient perturbation since it is applicable to all machine learning algorithms in general and is widely used for deep learning with differential privacy.

The amount of noise that must be added depends on the sensitivity of the machine learning algorithm. For instance, consider logistic regression with ℓ_2 regularization penalty. The objective function is of the form:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-X_i^\top \theta y_i}) + \frac{\lambda}{2} \| \theta \|_2^2$$

Assume that the training features are bounded, $\|X_i\|_2 \leq 1$ and $y_i \in \{-1, 1\}$. Chaudhuri et al. [12] prove that for this setting, objective perturbation requires sampling noise in the scale of $\frac{2}{n\epsilon}$, and output perturbation requires sampling noise in the scale of $\frac{2}{n\lambda\epsilon}$. The gradient of the objective function is:

$$\nabla J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{-X_i y_i}{1 + e^{X_i^\top \theta y_i}} + \lambda\theta$$

which has a sensitivity of $\frac{2}{n}$. Thus, gradient perturbation requires sampling noise in the scale of $\frac{2}{n\epsilon}$ at each iteration.

Deep learning. Deep learning follows the same learning procedure as in Algorithm 1, but the objective function is non-convex. As a result, the sensitivity analysis methods of Chaudhuri et al. [12] do not hold as they require a strong convexity assumption. Hence, their output and objective perturbation methods are not applicable. An alternative approach is to replace the non-convex function with a convex polynomial function [55, 56], and then use the standard objective perturbation. This approach requires carefully designing convex polynomial functions that can approximate the non-convexity, which can still limit the model's learning capacity. Moreover, it would require a considerable change in the existing machine learning infrastructure.

A simpler and more popular approach is to add noise to the gradients. Application of gradient perturbation requires a bound on the gradient norm. Since the gradient norm can be unbounded in deep learning, gradient perturbation can be used after manually clipping the gradients at each iteration. As noted by Abadi et al. [1], norm clipping provides a sensitivity bound on the gradients which is required for generating noise in gradient perturbation.

2.3 Implementing Differential Privacy

This section surveys how differential privacy has been used in machine learning applications, with a particular focus on the compromises implementers have made to obtain satisfactory utility. While the effective privacy provided by differential privacy mechanisms depends crucially on the choice of privacy budget ϵ, setting the ϵ value is discretionary and higher privacy budgets provide better utility.

Some of the early data analytics works on frequent pattern mining [7, 41], decision trees [21], private record linkage [30]

	Perturbation	Data Set	n	d	ϵ
Chaudhuri et al. [12]	Output and Objective	Adult	45,220	105	0.2
		KDDCup99	70,000	119	0.2
Pathak et al. [54]	Output	Adult	45,220	105	0.2
Hamm et al. [25]	Output	KDDCup99	493,000	123	1.0
		URL	200,000	50	1.0
Zhang et al. [78]	Objective	US	370,000	14	0.8
		Brazil	190,000	14	0.8
Jain and Thakurta [33]	Objective	CoverType	500,000	54	0.5
		KDDCup2010	20,000	2M	0.5
Jain and Thakurta [34]	Output and Objective	URL	100,000	20M	0.1
		COD-RNA	60,000	8	0.1
Song et al. [63]	Gradient	KDDCup99	50,000	9	1.0
		MNIST†	60,000	15	1.0
Wu et al. [70]	Output	Protein	72,876	74	0.05
		CoverType	498,010	54	0.05
Jayaraman et al. [35]	Output	Adult	45,220	104	0.5
		KDDCup99	70,000	122	0.5

Table 2: Simple ERM Methods which achieve High Utility with Low Privacy Budget.
† While MNIST is normally a 10-class task, Song et al. [63] use this for '1 vs rest' binary classification.

and recommender systems [47] were able to achieve both high utility and privacy with ϵ settings close to 1. These methods rely on finding frequency counts as a sub-routine, and hence provide ϵ-differential privacy by either perturbing the counts using Laplace noise or by releasing the top frequency counts using the exponential mechanism [48]. Machine learning, on the other hand, performs much more complex data analysis, and hence requires higher privacy budgets to maintain utility.

Next, we cover simple binary classification works that use small privacy budgets ($\epsilon \leq 1$). Then we survey complex classification tasks which seem to require large privacy budgets. Finally, we summarize recent works that aim to perform complex tasks with low privacy budgets by using relaxed definitions of differential privacy.

Binary classification. The first practical implementation of a private machine learning algorithm was proposed by Chaudhuri and Monteleoni [11]. They provide a novel sensitivity analysis under strong convexity constraints, allowing them to use output and objective perturbation for binary logistic regression. Chaudhuri et al. [12] subsequently generalized this method for ERM algorithms. This sensitivity analysis method has since been used by many works for binary classification tasks under different learning settings (listed in Table 2). While these applications can be implemented with low privacy budgets ($\epsilon \leq 1$), they only perform learning in restricted settings such as learning with low dimensional data, smooth objective functions and strong convexity assumptions, and are only applicable to simple binary classification tasks.

There has also been considerable progress in generalizing privacy-preserving machine learning to more complex scenarios such as learning in high-dimensional settings [33, 34, 64], learning without strong convexity assumptions [65], or relaxing the assumptions on data and objective functions [62, 68, 77]. However, these advances are mainly of theoretical interest and only a few works provide implementations [33, 34].

Complex learning tasks. All of the above works are limited to convex learning problems with binary classification tasks. Adopting their approaches to more complex learning tasks requires higher privacy budgets (see Table 3). For instance, the online version of ERM as considered by Jain et al. [32] requires ϵ as high as 10 to achieve acceptable utility. From the definition of differential privacy, we can see that $\Pr[\mathcal{M}(D) \in S] \leq e^{10} \times \Pr[\mathcal{M}(D') \in S]$. In other words, even if the model's output probability is 0.0001 on a data set D' that doesn't contain the target record, the model's output probability can be as high as 0.9999 on a neighboring data set D that contains the record. This allows an adversary to infer the presence or absence of a target record from the training data with high confidence. Adopting these binary classification methods for multi-class classification tasks requires even higher ϵ values. As noted by Wu et al. [70], it would require training a separate binary classifier for each class. Finally, high privacy budgets are required for non-convex learning algorithms, such as deep learning [60, 79]. Since the output and objective perturbation methods of Chaudhuri et al. [12] are not applicable to non-convex settings, implementations of differentially private deep learning rely on gradient pertur-

	Task	Perturbation	Data Set	n	d	C	ϵ
Jain et al. [32]	Online ERM	Objective	Year	500,000	90	2	10
			CoverType	581,012	54	2	10
Iyengar et al. [31]	Binary ERM	Objective	Adult	45,220	104	2	10
	Binary ERM		KDDCup99	70,000	114	2	10
	Multi-Class ERM		CoverType	581,012	54	7	10
	Multi-Class ERM		MNIST	65,000	784	10	10
	High Dimensional ERM		Gisette	6,000	5,000	2	10
Phan et al. [55, 56]	Deep Learning	Objective	YesiWell	254	30	2	1
			MNIST	60,000	784	10	1
Shokri and Shmatikov [60]	Deep Learning	Gradient	MNIST	60,000	1,024	10	369,200
			SVHN	100,000	3,072	10	369,200
Zhao et al. [79]	Deep Learning	Gradient	US	500,000	20	2	100
			MNIST	60,000	784	10	100

Table 3: Classification Methods for Complex Tasks

	Task	DP Relaxation	Data Set	n	d	C	ϵ
Huang et al. [28]	ERM	MA	Adult	21,000	14	2	0.5
Jayaraman et al. [35]	ERM	zCDP	Adult	45,220	104	2	0.5
			KDDCup99	70,000	122	2	0.5
Park et al. [53]	ERM	zCDP and MA	Stroke	50,345	100	2	0.5
			LifeScience	26,733	10	2	2.0
			Gowalla	1,256,384	2	2	0.01
			OlivettiFace	400	4,096	2	0.3
Lee [39]	ERM	zCDP	Adult	48,842	124	2	1.6
			US	40,000	58	2	1.6
			Brazil	38,000	53	2	1.6
Geumlek et al. [23]	ERM	RDP	Abalone	2,784	9	2	1.0
			Adult	32,561	100	2	0.05
			MNIST	7,988	784	2	0.14
Beaulieu et al. [6]	Deep Learning	MA	eICU	4,328	11	2	3.84
			TCGA	994	500	2	6.11
Abadi et al. [1]	Deep Learning	MA	MNIST	60,000	784	10	2.0
			CIFAR	60,000	3,072	10	8.0
Yu et al. [75]	Deep Learning	MA	MNIST	60,000	784	10	21.5
			CIFAR	60,000	3,072	10	21.5
Papernot et al. [52]	Deep Learning	MA	MNIST	60,000	784	10	2.0
			SVHN	60,000	3,072	10	8.0
Geyer et al. [24]	Deep Learning	MA	MNIST	60,000	784	10	8.0
Bhowmick et al. [8]	Deep Learning	MA	MNIST	60,000	784	10	3.0
			CIFAR	60,000	3,072	10	3.0
Hynes et al. [29]	Deep Learning	MA	CIFAR	50,000	3,072	10	4.0

Table 4: Gradient Perturbation based Classification Methods using Relaxed Notion of Differential Privacy

bation in their iterative learning procedure. These methods do not scale to large numbers of training iterations due to the composition theorem of differential privacy which causes the privacy budget to accumulate across iterations. The only exceptions are the works of Phan et al. [55, 56] that replace the non-linear functions in deep learning with polynomial approximations and then apply objective perturbation. With this transformation, they achieve high model utility for $\epsilon = 1$, as shown in Table 3. However, we note that this polynomial approximation is a non-standard approach to deep learning which can limit the model's learning capacity, and thereby diminish the model's accuracy for complex tasks.

Machine learning with relaxed DP definitions. To avoid the stringent composition property of differential privacy, several proposed privacy-preserving deep learning methods adopt the relaxed privacy definitions introduced in Section 2.1. Table 4 lists works that use gradient perturbation with relaxed notions of differential to reduce the overall privacy budget during iterative learning. The utility benefit of using relaxation is evident from the fact that the privacy budget for deep learning algorithms is significantly less than the prior works of Shokri and Shmatikov [60] and Zhao et al. [79] which do not use any relaxation.

While these *relaxed* definitions of differential privacy make complex iterative learning feasible for reasonable ϵ values, they might lead to more privacy leakage in practice. The main goal of our study is to evaluate the impact of implementation decisions regarding the privacy budget and relaxed definitions of differential privacy on the concrete privacy leakage that can be exploited by an attacker in practice. We do this by experimenting with various inference attacks, described in the next section.

3 Inference Attacks on Machine Learning

This section surveys the two types of inference attacks, *membership inference* (Section 3.1) and *attribute inference* (Section 3.2), and explains why they are useful metrics for evaluating privacy leakage. Section 3.3 briefly summarizes other relevant privacy attacks on machine learning.

3.1 Membership Inference

The aim of a *membership inference* attack is to infer whether or not a given record is present in the training set. Membership inference attacks can uncover highly sensitive information from training data. An early membership inference attack showed that it is possible to identify individuals contributing DNA to studies that analyze a mixture of DNA from many individuals, using a statistical distance measure to determine if a known individual is in the mixture [27].

Membership inference attacks can either be completely black-box where an attacker only has query access to the target model [61], or can assume that the attacker has full white-box access to the target model, along with some auxiliary information [74]. The first membership inference attack on machine learning was proposed by Shokri et al. [61]. They consider an attacker who can query the target model in a black-box way to obtain confidence scores for the queried input. The attacker tries to exploit the confidence score to determine whether the query input was present in the training data. Their attack method involves first training shadow models on a labelled data set, which can be generated either via black-box queries to the target model or through assumptions about the underlying distribution of training set. The attacker then trains an attack model using the shadow models to distinguish whether or not an input record is in the shadow training set. Finally, the attacker makes API calls to the target model to obtain confidence scores for each given input record and infers whether or not the input was part of the target model's training set. The inference model distinguishes the target model's predictions for inputs that are in its training set from those it did not train on. The key assumption is that the confidence score of the target model is higher for the training instances than it would be for arbitrary instances not present in the training set. This can be due to the generalization gap, which is prominent in models that overfit to training data.

A more targeted approach was proposed by Long et al. [44] where the shadow models are trained with and without a targeted input record t. At inference time, the attacker can check if the input record t was present in the training set of target model. This approach tests the membership of a specific record more accurately than Shokri et al.'s approach [61]. Recently, Salem et al. [59] proposed more generic membership inference attacks by relaxing the requirements of Shokri et al. [61]. In particular, requirements on the number of shadow models, knowledge of training data distribution and the target model architecture can be relaxed without substantially degrading the effectiveness of the attack.

Yeom et al. [74] recently proposed a more computationally efficient membership inference attack when the attacker has access to the target model and knows the average training loss of the model. To test the membership of an input record, the attacker evaluates the loss of the model on the input record and then classifies it as a member if the loss is smaller than the average training loss.

Connection to Differential Privacy. Differential privacy, by definition, aims to obfuscate the presence or absence of a record in the data set. On the other hand, membership inference attacks aim to identify the presence or absence of a record in the data set. Thus, intuitively these two notions counteract each other. Li et al. [42] point to this fact and provide a direct relationship between differential privacy and membership inference attacks. Backes et al. [4] studied membership inference attacks on microRNA studies and showed that differential privacy can reduce the success of membership

inference attacks, but at the cost of utility.

Yeom et al. [74] formally define a membership inference attack as an adversarial game where a data element is selected from the distribution, which is randomly either included in the training set or not. Then, an adversary with access to the trained model attempts to determine if that element was used in training. The *membership advantage* is defined as the difference between the adversary's true and false positive rates for this game. The authors prove that if the learning algorithm satisfies ϵ-differential privacy, then the adversary's advantage is bounded by $e^{\epsilon} - 1$. Hence, it is natural to use membership inference attacks as a metric to evaluate the privacy leakage of differentially private algorithms.

3.2 Attribute Inference

The aim of an *attribute inference* attack (also called *model inversion*) is to learn hidden sensitive attributes of a test input given at least API access to the model and information about the non-sensitive attributes. Fredrikson et al. [20] formalize this attack in terms of maximizing the posterior probability estimate of the sensitive attribute. More concretely, for a test record x where the attacker knows the values of its non-sensitive attributes $x_1, x_2, \cdots x_{d-1}$ and all the prior probabilities of the attributes, the attacker obtains the output of the model, $f(x)$, and attempts to recover the value of the sensitive attribute x_d. The attacker essentially searches for the value of x_d that maximizes the posterior probability $P(x_d \mid x_1, x_2, \cdots x_{d-1}, f(x))$. The success of this attack is based on the correlation between the sensitive attribute, x_d, and the model output, $f(x)$.

Yeom et al. [74] also propose an attribute inference attack using the same principle they use for their membership inference attack. The attacker evaluates the model's empirical loss on the input instance for different values of the sensitive attribute, and reports the value which has the maximum posterior probability of achieving the empirical loss. The authors define the *attribute advantage* similarly to their definition of membership advantage for membership inference.

Fredrikson et al. [20] demonstrated attribute inference attacks that could identify genetic markers based on warfarin dosage output by a model with just black-box access to model API.[1] With additional access to confidence scores of the model (noted as white-box information by Wu et al. [69]), more complex tasks have been performed, such as recovering faces from the training data [19].

Connection to Differential Privacy. Differential privacy is mainly tailored to obfuscate the presence or absence of a record in a data set, by limiting the effect of any single record on the output of differential private model trained on the data set. Logically this definition also extends to attributes or features of a record. In other words, by adding sufficient differential privacy noise, we should be able to limit the effect of a sensitive attribute on the model's output. This relationship between records and attributes is discussed by Yeom et al. [74]. Hence, we include these attacks in our experiments.

[1] This application has stirred some controversy based on the warfarin dosage output by the model itself being sensitive information correlated to the sensitive genetic markers, hence the assumption on attacker's prior knowledge of warfarin dosage is somewhat unrealistic [46].

3.3 Other Attacks on Machine Learning

Apart from inference attacks, many other attacks have been proposed in the literature which try to infer specific information from the target model. The most relevant are memorization attacks, which try to exploit the ability of high capacity models to memorize certain sensitive patterns in the training data [10]. These attacks have been found to be thwarted by differential privacy mechanisms with very little noise ($\epsilon = 10^9$) [10].

Other privacy attacks include model stealing, hyperparameter stealing, and property inference attacks. A model stealing attack aims to recover the model parameters via black-box access to the target model, either by adversarial learning [45] or by equation solving attacks [66]. Hyperparameter stealing attacks try to recover the underlying hyperparameters used during the model training, such as regularization coefficient [67] or model architecture [72]. These hyperparameters are intellectual property of commercial organizations that deploy machine learning models as a service, and hence these attacks are regarded as a threat to valuable intellectual property. A property inference attack tries to infer whether the training data set has a specific property, given a white-box access to the trained model. For instance, given access to a speech recognition model, an attacker can infer if the training data set contains speakers with a certain accent. Here the attacker can use the shadow training method of Shokri et al. [61] for distinguishing the presence and absence of a target property. These attacks have been performed on HMM and SVM models [3] and neural networks [22].

Though all these attacks may leak sensitive information about the target model or training data, the information leaked tends to be application-specific and is not clearly defined in a general way. For example, a property inference attack leaks some statistical property of the training data that is surprising to the model developer. Of course, the overall purpose of the model is to learn statistical properties from the training data. So, there is no general definition of a property inference attack without a prescriptive decision about which statistical properties of the training data should be captured by the model and which are sensitive to leak. In addition, the attacks mentioned in this section do not closely follow the threat model of differential privacy. Thus, we only consider inference attacks for our experimental evaluation.

In addition to these attacks, several poisoning and adversarial training attacks have been proposed [5, 50, 71, 73] which require an adversary that can actively interfere with the model

training process. We consider these out of scope for this paper, and assume a clean training process not under the control of the adversary.

4 Empirical Evaluation

To quantify the privacy leakage of the differentially private implementations for machine learning, we conduct experiments to measure how much an adversary can infer from a model. As motivated in Section 3, we measure privacy leakage using membership and attribute inference in our experiments. Note, however, that the conclusions we can draw from experiments like this are limited to showing a lower bound on the information leakage since they are measuring the effectiveness of a particular attack. Such experimental results cannot be used to make strong claims about what the best possible attack would be able to infer, especially in cases where an adversary has auxiliary information to help guide the attack. Evidence from our experiments, however, does provide clear evidence for when implemented privacy protections do not appear to provide sufficient privacy.

4.1 Experimental Setup

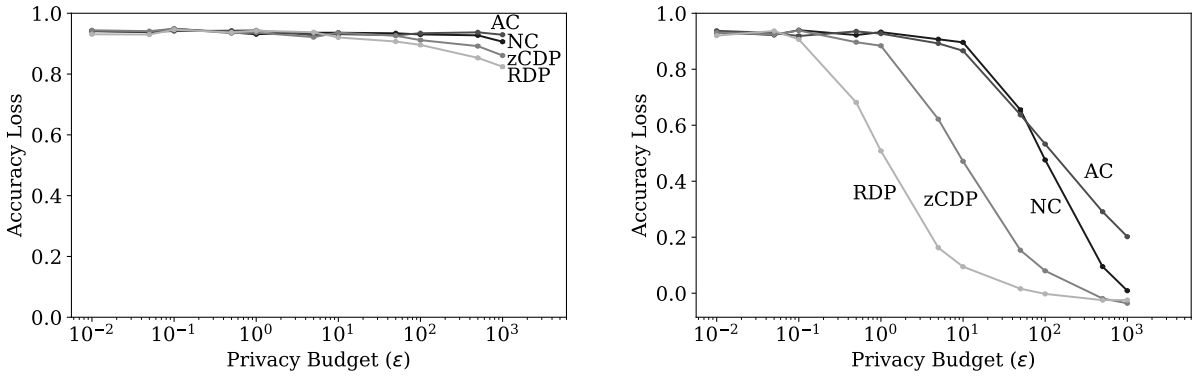
We evaluate the privacy leakage of two differentially private algorithms using gradient perturbation: logistic regression for empirical risk minimization (Section 4.2) and neural networks for non-convex learning (Section 4.3). For both, we consider the different relaxed notions of differential privacy and compare their privacy leakage. The variations that we implement are naïve composition (NC), advanced composition (AC), zero-concentrated differential privacy (zCDP) and Rényi differential privacy (RDP) (see Section 2.1 for details). We do not include CDP as it has the same composition property as zCDP (Table 1). For RDP, we use the RDP accountant (previously moments accountant) of TF Privacy framework [2].

We evaluate the models on two main metrics: *accuracy loss*, the model's accuracy loss on test set with respect to the non-private baseline, and *privacy leakage*, the attacker's advantage as defined by Yeom et al. [74]. To evaluate out the inference attack, we provide the attacker with a set of 20,000 records consisting of 10,000 records from training set and 10,000 records from the test set. We call records in the training set *members*, and the other records *non-members*. These labels are not known to the attacker. The task of the attacker is to predict whether or not a given input record belongs to the training set (i.e., if it is a member). The privacy leakage metric is calculated by taking the difference between the true positive rate (TPR) and the false positive rate (FPR) of the inference attack. Thus the privacy leakage metric is always between 0 and 1, where the value of 0 indicates that there is no leakage. For example, if an attacker performs membership inference on a model and obtains a privacy leakage of 0.7 then it implies that for every 100 wrong membership predictions made by the attacker, 170 'true' members are revealed to the attacker. In other words, 170 training records are revealed to the attacker. To better understand the potential impact of leakage, we also conduct experiments to estimate the actual number of members who are at risk for disclosure in a membership inference attack.

Data sets. We evaluate our models over two data sets for multi-class classification tasks: CIFAR-100 [38] and Purchase-100 [36]. CIFAR-100 consists of 28×28 images of 100 real world objects, with 500 instances of each object class. We use PCA to reduce the dimensionality of records to 50. The Purchase-100 data set consists of 200,000 customer purchase records of size 100 each (corresponding to the 100 frequently-purchased items) where the records are grouped into 100 classes based on the customers' purchase style. For both data sets, we use 10,000 randomly-selected instances for training and 10,000 randomly-selected non-training instances for the test set. The remaining records are used for training shadow models and inference model.

Attacks. For our experiments, we use the attack frameworks of Shokri et al. [61] and Yeom et al. [74] for membership inference and the method proposed by Yeom et al. [74] for attribute inference. In Shokri et al.'s framework [61], multiple shadow models are trained on data that is sampled from the same distribution as the private data set. These shadow models are used to train an inference model to identify whether an input record belongs to the private data set. The inference model is trained using a set of records used to train the shadow models, a set of records randomly selected from the distribution that are not part of the shadow model training, along with the confidence scores output by the shadow models for all of the input records. Using these inputs, the inference model learns to distinguish the training records from the non-training records. At the inference stage, the inference model takes an input record along with the confidence score of the target model on the input record, and outputs whether the input record belongs to the target model's private training data set. The intuition is that if the target model overfits on its training set, its confidence score for a training record will be higher than its confidence score for an otherwise similar input that was not used in training. The inference model tries to exploit this property. In our instantiation of the attack framework, we use five shadow models which all have the same model architecture as the target model. Our inference model is a neural network with two hidden layers of size 64. This setting is consistent with the original work [61].

The attack framework of Yeom et al. [74] is simpler than Shokri et al.'s design. It assumes a white-box attacker with access to the target model's expected training loss on the private data set, in addition to having access to the target model. For membership inference, the attacker simply observes the target model's loss on the input record. The attacker classifies the record as a member if the loss is smaller than the target



(a) Batch gradient clipping

(b) Per-instance gradient clipping

Figure 1: Impact of clipping on accuracy loss of logistic regression (CIFAR-100).

model's expected training loss, otherwise the record is classified as a non-member. The same principle is used for attribute inference. Given an input record, the attacker brute-forces all possible values for the unknown private attribute and observes the target model's loss, outputting the value for which the loss is closest to the target's expected training loss. Since there are no attributes in our data sets that are explicitly annotated as private, we randomly choose five attributes, and perform the attribute inference attack on each attribute independently, and report the averaged results.

Hyperparameters. For both data sets, we train logistic regression and neural network models with ℓ_2 regularization. First, we train a non-private model and perform a grid search over the regularization coefficient λ to find the value that minimizes the classification error on the test set. For CIFAR-100, we found optimal values to be $\lambda = 10^{-5}$ for logistic regression and $\lambda = 10^{-4}$ for neural network. For Purchase-100, we found optimal values to be $\lambda = 10^{-5}$ for logistic regression and $\lambda = 10^{-8}$ for neural network. Next, we fix this setting to train differentially private models using gradient perturbation. We vary ϵ between 0.01 and 1000 while keeping $\delta = 10^{-5}$, and report the accuracy loss and privacy leakage. The choice of $\delta = 10^{-5}$ satisfies the requirement that δ should be smaller than the inverse of the training set size 10,000. We use the ADAM optimizer for training and fix the learning rate to 0.01 with a batch size of 200. Due to the random noise addition, all the experiments are repeated five times and the average results and standard errors are reported. We do not assume pre-trained model parameters, unlike the prior works of Abadi et al. [1] and Yu et al. [75].

Clipping. For gradient perturbation, clipping is required to bound the sensitivity of the gradients. We tried clipping at both the batch and per-instance level. Batch clipping is more computationally efficient and a standard practice in deep learning. On the other hand, per-instance clipping uses the privacy budget more efficiently, resulting in more accurate models for a given privacy budget. We use the TensorFlow Privacy framework [2] which implements both batch and per-instance clipping. We fix the clipping threshold at $C = 1$.

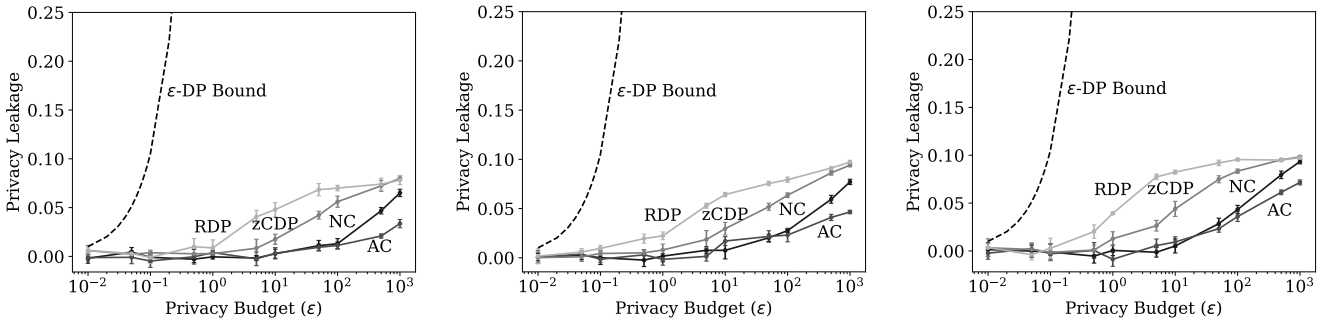
Figure 1 compares the accuracy loss of logistic regression models trained over CIFAR-100 data set with both batch clipping and per-instance clipping. Per-instance clipping allows learning more accurate models for all values of ϵ and amplifies the differences between the different mechanisms. For example, the model trained with RDP achieves accuracy close to the non-private model for $\epsilon = 100$ when performing per-instance clipping. Whereas, the models do not learn anything useful when using batch clipping. Hence, for the rest of the paper we only report the results for per-instance clipping.

4.2 Logistic Regression Results

We train ℓ_2-regularized logistic regression models on both the CIFAR-100 and Purchase-100 data sets.

CIFAR-100. The baseline model for non-private logistic regression achieves accuracy of 0.225 on training set and 0.155 on test set, which is competitive with the state-of-art neural network model [61] that achieves test accuracy close to 0.20 on CIFAR-100 after training on larger data set. Thus, there is a small generalization gap of 0.07, which the inference attacks try to exploit.

Figure 1(b) compares the accuracy loss for logistic regression models trained with different relaxed notions of differential privacy as we varying the privacy budget ϵ. The accuracy loss is normalized with respect to the accuracy of non-private model to clearly depict the model utility. An accuracy loss value of 1 means that the model has 100% loss and hence has no utility, whereas the value of 0 means that the model achieves same accuracy as the non-private baseline. As depicted in the figure, naïve composition achieves accuracy



(a) Shokri et al. membership inference (b) Yeom et al. membership inference (c) Yeom et al. attribute inference

Figure 2: Inference attacks on logistic regression (CIFAR-100).

close to 0.01 for $\epsilon \leq 10$ which is random guessing for 100-class classification. Naïve composition achieves accuracy loss close to 0 for $\epsilon = 1000$. Advanced composition adds more noise than naïve composition when privacy budget is greater than the number of training epochs ($\epsilon \geq 100$). The relaxations zCDP and RDP achieve accuracy loss close to 0 at $\epsilon = 500$ and $\epsilon = 50$ respectively, which is order of magnitudes smaller than the naïve composition. This is expected since the relaxed definitions require less added noise.

Figures 2(a) and 2(b) show the privacy leakage due to membership inference attacks on logistic regression models. Figure 2(a) shows results for the black-box attacker of Shokri et al. [61], which has access to the target model's confidence scores on the input record. Naïve composition achieves privacy leakage close to 0 for $\epsilon \leq 10$, and the leakage reaches 0.065 ± 0.004 for $\epsilon = 1000$. The relaxed variants RDP and zCDP have average leakage close to 0.080 ± 0.004 for $\epsilon = 1000$. As expected, the differential privacy variations have leakage in accordance with the amount of noise they add for a given ϵ. The plots also show the theoretical upper bound on the privacy leakage for ϵ-differential privacy, where the bound is $e^{\epsilon} - 1$ (see Section 3.1).

Figure 2(b) shows results for the white-box attacker of Yeom et al. [61], which has access to the target model's loss on the input record. As expected, zCDP and RDP relaxations leak the most. Naïve composition does not have any significant leakage for $\epsilon \leq 10$, but the leakage reaches 0.077 ± 0.003 for $\epsilon = 1000$. The observed leakage of all the variations is in accordance with the noise magnitude required for different differential privacy guarantees.

Figure 2(c) depicts the privacy leakage due to the attribute inference attack. The privacy leakage of RDP is highest, closely followed by zCDP. Naïve composition has low privacy leakage for $\epsilon \leq 10$ (attacker advantage of 0.005 ± 0.007 at $\epsilon = 10$), but it quickly increases to 0.093 ± 0.002 for $\epsilon = 1000$. But for meaningful privacy budgets, there is no significant leakage (< 0.02) for any of the methods. As expected, across all variations as privacy budgets increase both the attacker's advantage (privacy leakage) and the model utility (accuracy) increase. For this example, there is no choice of ϵ available that provides any effective privacy for a model that does better than random guessing.

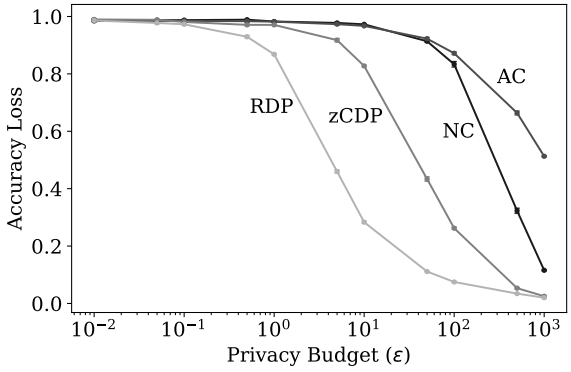


Figure 3: Accuracy loss of logistic regression (Purchase-100).

To gain more understanding of the impact of privacy leakage, Table 5 shows the actual number of training set members exposed to the attacker for different differential privacy variations. We assume the attacker has some limited tolerance for falsely exposing a member (that is, a bound on the acceptable false positive rate), and sets the required threshold score for the inference model output as the level needed to achieve that false positive rate. Then, we count the number of members in the private training data set for whom the inference model output exceeds that confidence threshold. Table 5 reports the number of members exposed to an adversary who tolerates false positive rates of 1%, 2%, and 5%. As we increase the tolerance threshold, there is a gradual increase in membership leakage for all the methods, and the leakage of relaxed variants increases drastically. Naïve composition and advanced composition are resistant to attack for $\epsilon \leq 10$, whereas zCDP is resistant to attack for $\epsilon \leq 1$. RDP is resistant up to $\epsilon = 0.05$.

	Naïve Composition				Advanced Composition				zCDP				RDP			
ϵ	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%
0.01	.93	0	0	0	.94	0	0	0	.93	0	0	0	.92	0	0	0
0.05	.92	0	0	0	.93	0	0	0	.92	0	0	0	.94	0	0	0
0.1	.94	0	0	0	.92	0	0	0	.94	0	0	0	.91	0	0	1
0.5	.92	0	0	0	.94	0	0	0	.90	0	0	0	.68	0	3	27
1.0	.93	0	0	0	.93	0	0	0	.88	0	0	0	.51	4	21	122
5.0	.91	0	0	0	.89	0	0	0	.62	2	11	45	.16	39	95	304
10.0	.90	0	0	0	.87	0	0	0	.47	15	38	137	.09	55	109	329
50.0	.65	0	2	16	.64	19	31	73	.15	44	102	291	.02	70	142	445
100.0	.48	6	29	152	.53	18	47	138	.08	58	121	362	.00	76	158	456
500.0	.10	53	112	328	.29	42	88	256	.00	80	159	487	.00	86	166	516
1,000.0	.01	65	138	413	.20	57	111	301	.00	86	172	514	.00	93	185	530

Table 5: Number of individuals (out of 10,000) exposed by Yeom et al. membership inference attack on logistic regression (CIFAR-100). The non-private ($\epsilon = \infty$) model leaks 129, 240 and 704 members for 1%, 2% and 5% FPR respectively.

Purchase-100. The baseline model for non-private logistic regression achieves accuracy of 0.942 on the training set and 0.695 on test set. In comparison, Google ML platform's black-box trained model achieves a test accuracy of 0.656 for Purchase-100 (see Shokri et al. [61] for details).

Figure 3 shows the accuracy loss of all differential privacy variants on Purchase-100 data set. Naïve composition and advanced composition have essentially no utility until ϵ exceeds 100. At $\epsilon = 1000$, naïve composition achieves accuracy loss of 0.116 ± 0.003, the advanced composition achieves accuracy loss of 0.513 ± 0.003 and the other variants achieve accuracy loss close to 0.02. RDP achieves the best utility across all ϵ values. zCDP performs better than advanced composition and naïve composition.

Figure 4 compares the privacy leakage of the variants against the inference attacks. The leakage is in accordance to the noise each variant adds and it increases proportionally to the model utility. Hence, if a model has reasonable utility, it is bound to leak membership information. The white-box membership inference attack of Yeom et al. is relatively more effective than the black-box membership inference attack of Shokri et al. as shown in Figures 4(a) and 4(b). Table 6 shows the number of individual members exposed, with similar results to the findings for CIFAR-100.

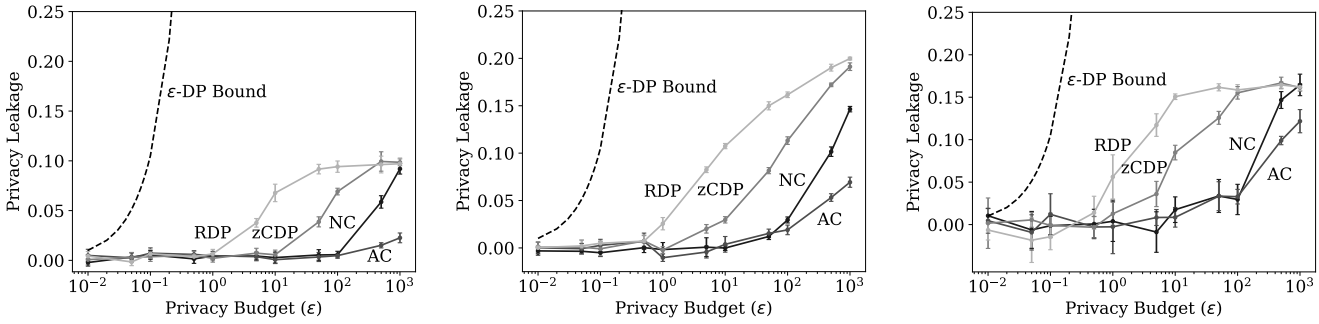
4.3 Neural Networks

We train a neural network model consisting of two hidden layers and an output layer. The hidden layers have 256 neurons that use ReLU activation. The output layer is a softmax layer with 100 neurons, each corresponding to a class label. This architecture is similar to the one used by Shokri et al. [61].

CIFAR-100. The baseline non-private neural network model achieves accuracy of 1.000 on the training set and 0.168 on test set, which is competitive to the neural network model of Shokri et al. [61]. Their model is trained on a training set of size 29,540 and achieves test accuracy of 0.20, whereas our model is trained on 10,000 training instances. There is a huge generalization gap of 0.832, which the inference attacks can exploit. Figure 5(a) compares the accuracy loss of neural network models trained with different relaxed notions of differential privacy with varying privacy budget ϵ. The model trained with naïve composition does not learn anything useful until $\epsilon = 100$ (accuracy loss of 0.907 ± 0.004), at which point the advanced composition also has accuracy loss close to 0.935 and the other variants achieve accuracy loss close to 0.24. None of the variants approach zero accuracy loss, even for $\epsilon = 1000$. The relative performance is similar to that of logistic regression model discussed in Section 4.2.

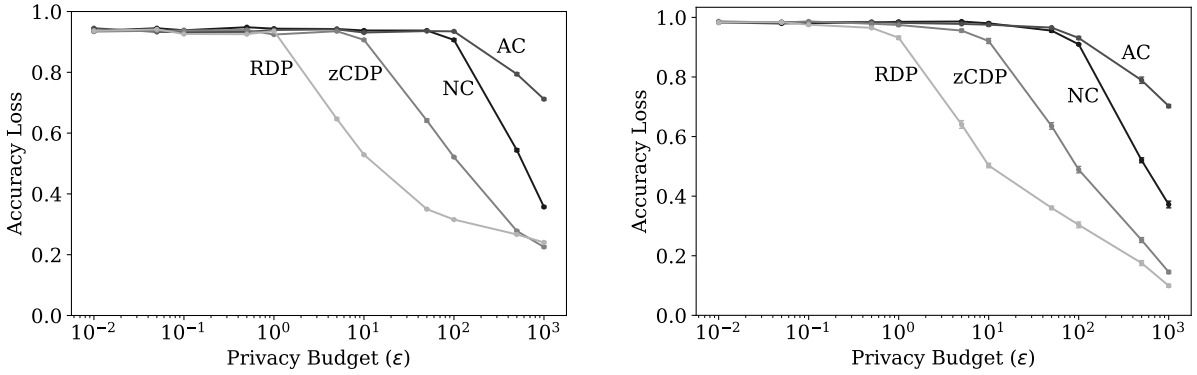
Figures 6(a) and 6(b) shows the privacy leakage due to membership inference attacks on neural network models trained with different relaxed notions for both attacks. The privacy leakage for each variation of differential privacy accords with the amount of noise it adds to the model. The leakage is significant for relaxed variants at higher ϵ values due to model overfitting. For $\epsilon = 1000$, with the Shokri et al. attack, naïve composition has leakage of 0.034 compared to 0.002 for advanced composition, 0.219 for zCDP, and 0.277 for RDP (above the region shown in the plot). For the white-box attacker of Yeom et al. [74], RDP leaks the most for $\epsilon = 1000$ (membership advantage of 0.399) closely followed by zCDP. This is because these relaxed variations add considerably less noise in comparison to naïve composition. Naïve composition and advanced composition achieve strong privacy against membership inference attackers, but fail to learning anything useful. No option appears to provide both acceptable model utility and meaningful privacy.

Like we did for logistic regression, we report the actual number of training set members exposed to the attacker in Table 7. The impact of privacy leakage is far more severe for the non-private neural network model due to model overfitting—



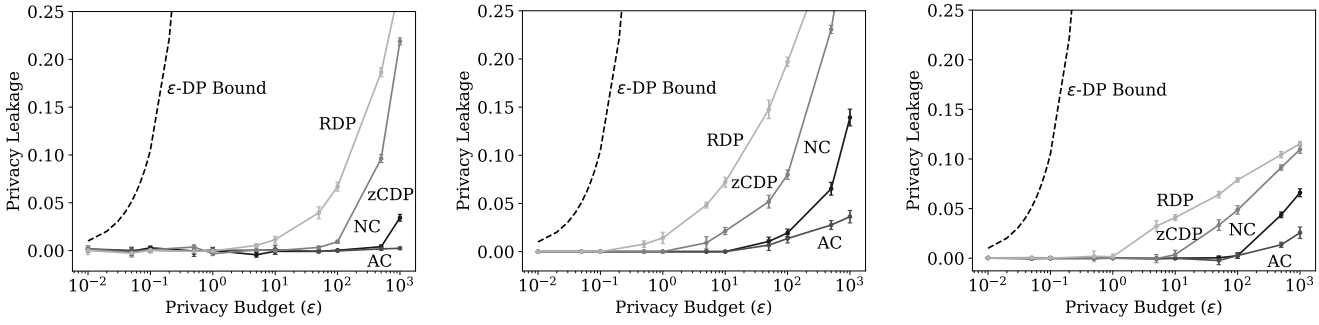
(a) Shokri et al. membership inference (b) Yeom et al. membership inference (c) Yeom et al. attribute inference

Figure 4: Inference attacks on logistic regression (Purchase-100).



(a) CIFAR-100 (b) Purchase-100

Figure 5: Accuracy loss of neural networks.



(a) Shokri et al. membership inference (b) Yeom et al. membership inference (c) Yeom et al. attribute inference

Figure 6: Inference attacks on neural network (CIFAR-100).

	Naïve Composition				Advanced Composition				zCDP				RDP			
ϵ	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%
0.01	.98	0	0	0	.99	0	0	0	.99	0	0	0	.99	0	0	0
0.05	.99	0	0	0	.98	0	0	0	.99	0	0	0	.98	0	0	0
0.1	.99	0	0	0	.98	0	0	0	.98	0	0	0	.97	0	0	0
0.5	.98	0	0	0	.98	0	0	0	.97	0	0	0	.93	0	0	2
1.0	.98	0	0	0	.98	0	0	0	.97	0	0	0	.87	0	0	23
5.0	.98	0	0	0	.97	0	0	0	.92	0	0	4	.46	42	72	174
10.0	.97	0	0	0	.97	0	0	0	.83	1	7	35	.28	53	101	270
50.0	.91	0	0	1	.92	0	0	1	.43	38	65	187	.11	72	154	406
100.0	.83	0	0	28	.87	0	1	10	.26	55	113	289	.08	84	160	473
500.0	.32	45	95	227	.66	18	34	84	.05	77	183	487	.03	75	181	533
1,000.0	.12	81	164	427	.51	34	58	145	.02	87	184	530	.02	94	189	566

Table 6: Number of members (out of 10,000) exposed by Yeom et al. membership inference attack on logistic regression (Purchase-100). The non-private ($\epsilon = \infty$) model leaks 102, 262 and 716 members for 1%, 2% and 5% FPR respectively.

	Naïve Composition				Advanced Composition				zCDP				RDP			
ϵ	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%	Loss	1%	2%	5%
0.01	.94	0	0	0	.94	0	0	0	.93	0	0	0	.94	0	0	0
0.05	.94	0	0	0	.93	0	0	0	.94	0	0	0	.94	0	0	0
0.1	.94	0	0	0	.93	0	0	0	.94	0	0	0	.93	0	0	0
0.5	.95	0	0	0	.93	0	0	0	.94	0	0	0	.92	0	0	0
1.0	.94	0	0	0	.94	0	0	0	.92	0	0	0	.94	0	0	0
5.0	.94	0	0	0	.94	0	0	0	.94	0	0	0	.65	11	24	79
10.0	.94	0	0	0	.93	0	0	0	.91	0	0	2	.53	9	33	108
50.0	.94	0	0	0	.94	0	0	0	.64	2	12	65	.35	28	65	185
100.0	.91	0	0	0	.93	0	0	0	.52	13	31	98	.32	21	67	205
500.0	.54	3	21	58	.79	4	7	31	.28	8	41	210	.27	5	54	278
1,000.0	.36	20	48	131	.71	8	16	74	.22	12	42	211	.24	10	37	269

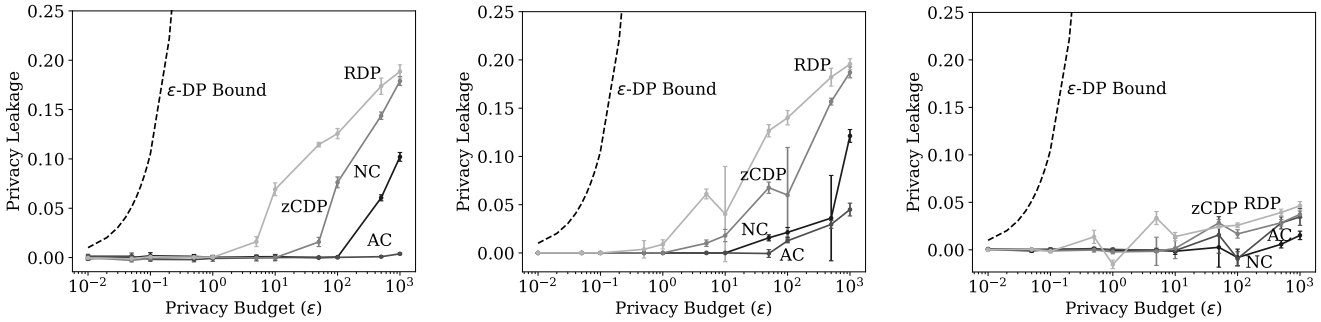
Table 7: Number of members (out of 10,000) exposed by Yeom et al. membership inference attack on neural network (CIFAR-100). The non-private ($\epsilon = \infty$) model leaks 0, 556 and 7349 members for 1%, 2% and 5% FPR respectively.

exposing over 73% of training set members at 5% false positive rate, compared to only 7% for the logistic regression model.[2] The privacy mechanisms provide substantial reduction in exposure, even with high ϵ budgets, but the relaxed variants expose more members compared to naïve composition and advanced composition.

Figure 6(c) depicts the privacy leakage due to attribute inference attack on the neural network models. Naïve composition and advanced composition are both resistant to the attack for $\epsilon \leq 100$, but the relaxed variants reveal some privacy leakage for lower privacy budgets.

Purchase-100. The baseline non-private neural network model achieves accuracy of 0.982 on the training set and 0.605 on test set. In comparison, the neural network model of Shokri et al. [61] trained on a similar data set (but with 600 attributes instead of 100 as in our data set) achieves 0.670 test accuracy. Figure 5(b) compares the accuracy loss, and Figure 7 the privacy leakage, of neural network models trained with different variants of differential privacy. The trends for both accuracy and privacy are similar to those for the logistic regression models (Figure 3). The relaxed variants achieve model utility close to the non-private baseline for $\epsilon = 1000$, while naïve composition continues to suffer from high accuracy loss (0.372). Advanced composition has higher accuracy loss of 0.702 for $\epsilon = 1000$ as it requires addition of more noise than naïve composition when ϵ is greater than the number of training epochs. Figure 7 shows the privacy leakage comparison of the variants against the inference attacks. The results are consistent with those observed for CIFAR-100.

[2]Curiously, this appears to be contradicted at 1% FPR where no members are revealed by non-private NN model but some are revealed by the privacy-preserving models. This is due to the number of extremely high-confidence incorrect outputs of the non-private model, meaning that there is no confidence threshold that does not include at least 1% false positives.



(a) Shokri et al. membership inference (b) Yeom et al. membership inference (c) Yeom et al. attribute inference

Figure 7: Inference attacks on neural network (Purchase-100).

4.4 Discussion

While the tighter cumulative noise bounds provided by relaxed variants of differential privacy improve model utility for a given privacy budget, the reduction in noise increases vulnerability to inference attacks. Thus, privacy does not come for free, and the relaxations of the differential privacy definition that result in lower noise requirements come with additional privacy risks. While these relaxed definitions still satisfy the (ϵ, δ)-differential privacy guarantees, the concrete value of these guarantees diminishes rapidly with high ϵ values and non-zero δ. Although the theoretical guarantees provided by differential privacy are very appealing, once ϵ values exceed small values, the practical value of these guarantees is insignificant—in most of our inference attack figures, the theoretical bound given by ϵ-DP falls off the graph before any measurable privacy leakage occurs (and at levels well before models provide acceptable utility). The value of these privacy mechanisms comes not from the theoretical guarantees, but from the impact of the mechanism on what realistic adversaries can infer.

We note that in our inference attack experiments, we use equal numbers of member and non-member records which provides 50-50 prior success probability to the attacker. Thus, even an ϵ-DP implementation might leak even for small ϵ values, though we did not observe any such leakage. Alternatively, a skewed prior probability may lead to smaller leakage even for large ϵ values. Our goal in this work is to evaluate scenarios where risk of inference is high, so the use of 50-50 prior probability is justified. We also emphasis that our results show the privacy leakage due to two particular membership inference attacks. Attacks only get better, so future attacks may be able to infer more than is shown in our experiments.

5 Conclusion

Differential privacy has earned a well-deserved reputation providing principled and powerful mechanisms for ensuring provable privacy. However, when it is implemented for challenging tasks such as machine learning, compromises must be made to preserve utility. It is essential that the privacy impact of those compromises is well understood when differential privacy is deployed to protect sensitive data. Our results are a step towards improving that understanding, and reveal that the commonly-used relaxations of differential privacy may provide unacceptable utility-privacy trade-offs. We hope our study will encourage more careful assessments of the practical privacy value of formal claims based on differential privacy, and lead to deeper understanding of the privacy impact of design decisions when deploying differential privacy, and eventually to solutions that provide desirable, and well understood, utility-privacy trade-offs.

Availability

Open source code for reproducing all of our experiments is available at https://github.com/bargavj/EvaluatingDPML.

Acknowledgments

The authors are deeply grateful to Úlfar Erlingsson for pointing out some key misunderstandings in an early version of this work and for convincing us of the importance of per-instance gradient clipping, and to Úlfar, Ilya Mironov, and Shuang Song for help validating and improving the work. We thank Vincent Bindschaedler for shepherding our paper. We thank Youssef Errami and Jonah Weissman for contributions to the experiments, and Ben Livshits for feedback on the work. Atallah Hezbor, Faysal Shezan, Tanmoy Sen, Max Naylor, Joshua Holtzman and Nan Yang helped systematize the related works. Finally, we thank Congzheng Song and Samuel Yeom for providing their implementation of inference attacks. This work was partially funded by grants from the National Science Foundation SaTC program (#1717950, #1915813) and support from Intel and Amazon.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security*, 2016.

[2] Galen Andrew, Steve Chien, and Nicolas Papernot. TensorFlow Privacy. https://github.com/tensorflow/privacy.

[3] Giuseppe Ateniese, Luigi Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 2015.

[4] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. Membership privacy in MicroRNA-based studies. In *ACM Conference on Computer and Communications Security*, 2016.

[5] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *arXiv:1807.00459*, 2018.

[6] Brett K Beaulieu-Jones, William Yuan, Samuel G Finlayson, and Zhiwei Steven Wu. Privacy-preserving distributed deep learning for clinical data. *arXiv:1812.01484*, 2018.

[7] Raghav Bhaskar, Srivatsan Laxman, Adam Smith, and Abhradeep Thakurta. Discovering frequent patterns in sensitive data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[8] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv:1812.00984*, 2018.

[9] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, 2016.

[10] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The Secret Sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

[11] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, 2009.

[12] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private Empirical Risk Minimization. *Journal of Machine Learning Research*, 2011.

[13] Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Detecting violations of differential privacy. In *ACM Conference on Computer and Communications Security*, 2018.

[14] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011.

[15] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Symposium on Foundations of Computer Science*, 2013.

[16] Cynthia Dwork. Differential Privacy: A Survey of Results. In *International Conference on Theory and Applications of Models of Computation*, 2008.

[17] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 2014.

[18] Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *arXiv:1603.01887*, 2016.

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM Conference on Computer and Communications Security*, 2015.

[20] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*.

[21] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[22] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *ACM Conference on Computer and Communications Security*, 2018.

[23] Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. Rényi differential privacy mechanisms for posterior sampling. In *Advances in Neural Information Processing Systems*, 2017.

[24] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv:1712.07557*, 2017.

[25] Jihun Hamm, Paul Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, 2016.

[26] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. Principled evaluation of differentially private algorithms using DPBench. In *ACM SIGMOD Conference on Management of Data*, 2016.

[27] Nils Homer et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures us-

ing high-density SNP genotyping microarrays. *PLoS Genetics*, 2008.

[28] Zonghao Huang, Rui Hu, Yanmin Gong, and Eric Chan-Tin. DP-ADMM: ADMM-based distributed learning with differential privacy. *arXiv:1808.10101*, 2018.

[29] Nick Hynes, Raymond Cheng, and Dawn Song. Efficient deep learning on multi-source private data. *arXiv:1807.06689*, 2018.

[30] Ali Inan, Murat Kantarcioglu, Gabriel Ghinita, and Elisa Bertino. Private record matching using differential privacy. In *International Conference on Extending Database Technology*, 2010.

[31] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *IEEE Symposium on Security and Privacy*, 2019.

[32] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Annual Conference on Learning Theory*, 2012.

[33] Prateek Jain and Abhradeep Thakurta. Differentially private learning with kernels. In *International Conference on Machine Learning*, 2013.

[34] Prateek Jain and Abhradeep Guha Thakurta. (Near) Dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, 2014.

[35] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving Empirical Risk Minimization. In *Advances in Neural Information Processing Systems*, 2018.

[36] Kaggle, Inc. Acquire Valued Shoppers Challenge. https://kaggle.com/c/acquire-valued-shoppers-challenge/data, 2014.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[38] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[39] Jaewoo Lee. Differentially private variance reduced stochastic gradient descent. In *International Conference on New Trends in Computing Sciences*, 2017.

[40] Dong-Hui Li and Masao Fukushima. A modified BFGS method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics*, 2001.

[41] Ninghui Li, Wahbeh Qardaji, Dong Su, and Jianneng Cao. PrivBasis: Frequent itemset mining with differential privacy. *The VLDB Journal*, 2012.

[42] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *ACM Conference on Computer and Communications Security*, 2013.

[43] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 1989.

[44] Yunhui Long, Vincent Bindschaedler, and Carl A. Gunter. Towards measuring membership privacy. *arXiv:1712.09136*, 2017.

[45] Daniel Lowd and Christopher Meek. Adversarial learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2005.

[46] Frank McSherry. Statistical inference considered harmful. https://github.com/frankmcsherry/blog/blob/master/posts/2016-06-14.md, 2016.

[47] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[48] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Symposium on Foundations of Computer Science*, 2007.

[49] Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium*, 2017.

[50] Luis Munoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

[51] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *ACM Symposium on Theory of Computing*, 2007.

[52] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

[53] Mijung Park, Jimmy Foulds, Kamalika Chaudhuri, and Max Welling. DP-EM: Differentially private expectation maximization. In *Artificial Intelligence and Statistics*, 2017.

[54] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty Differential Privacy via Aggregation of Locally

Trained Classifiers. In *Advances in Neural Information Processing Systems*, 2010.

[55] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. In *AAAI Conference on Artificial Intelligence*, 2016.

[56] NhatHai Phan, Xintao Wu, and Dejing Dou. Preserving differential privacy in convolutional deep belief networks. *Machine Learning*, 2017.

[57] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 1992.

[58] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 2018.

[59] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security Symposium*.

[60] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *ACM Conference on Computer and Communications Security*, 2015.

[61] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

[62] Adam Smith and Abhradeep Thakurta. Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso. In *Proceedings of Conference on Learning Theory*, 2013.

[63] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing*, 2013.

[64] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Private Empirical Risk Minimization beyond the worst case: The effect of the constraint set geometry. *arXiv:1411.5417*, 2014.

[65] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly Optimal Private LASSO. In *Advances in Neural Information Processing Systems*, 2015.

[66] Florian Tramèr, Fan Zhang, Ari Juels, Michael Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security Symposium*, 2016.

[67] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *IEEE Symposium on Security and Privacy*, 2018.

[68] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private Empirical Risk Minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, 2017.

[69] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. A methodology for formalizing model-inversion attacks. In *IEEE Computer Security Foundations Symposium*, 2016.

[70] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *ACM SIGMOD Conference on Management of Data*, 2017.

[71] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 2015.

[72] Mengjia Yan, Christopher Fletcher, and Josep Torrellas. Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. *arXiv:1808.04761*, 2018.

[73] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv:1703.01340*, 2017.

[74] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, 2018.

[75] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *IEEE Symposium on Security and Privacy*, 2019.

[76] Matthew D Zeiler. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*, 2012.

[77] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private ERM for smooth objectives. In *International Joint Conference on Artificial Intelligence*, 2017.

[78] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *The VLDB Journal*, 2012.

[79] Lingchen Zhao, Yan Zhang, Qian Wang, Yanjiao Chen, Cong Wang, and Qin Zou. Privacy-preserving collaborative deep learning with irregular participants. *arXiv:1812.10113*, 2018.