Bargav Jayaraman*, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans

Revisiting Membership Inference Under Realistic Assumptions

Abstract: We study membership inference in settings where assumptions commonly used in previous research are relaxed. First, we consider cases where only a small fraction of the candidate pool targeted by the adversary are members and develop a PPV-based metric suitable for this setting. This skewed prior setting is more realistic than the balanced prior setting typically considered. Second, we consider adversaries that select inference thresholds according to their attack goals, such as identifying as many members as possible with a given false positive tolerance. We develop a threshold selection designed for achieving particular attack goals. Since previous inference attacks fail in imbalanced prior settings, we develop new inference attacks based on the intuition that inputs corresponding to training set members will be near a local minimum in the loss function. An attack that combines this with thresholds on the per-instance loss can achieve high PPV even in settings where other attacks are ineffective.

DOI 10.2478/popets-2021-0031
Received 2020-08-31; revised 2020-12-15; accepted 2020-12-16.

1 Introduction

Differential privacy has become the gold standard for performing any privacy-preserving statistical analysis over sensitive data. Its privacy-utility tradeoff is controlled by the privacy loss budget parameter ϵ (and failure probability δ). While it is a well known fact that larger privacy loss budgets lead to more leakage, it is still an open question how low privacy loss budgets should be to provide meaningful privacy in practice.

Although differential privacy provides strong bounds on the worst-case privacy loss, it does not elucidate what privacy attacks could be realized in practice. Attacks, on the other hand, provide an empirical lower bound on privacy leakage for a particular setting. Many attacks on machine learning algorithms have been proposed that aim to infer private information about the model or the training data. These attacks include membership inference [29, 35, 38, 46], attribute inference [15, 16, 46], property inference [3, 17], model stealing [31, 42] and hyperparameter stealing [43, 45]. Of these, membership inference attacks are most directly connected to the differential privacy definition, and thus are a good basis for evaluating the privacy leakage of differentially private mechanisms. Given a small enough privacy loss budget, a differentially private mechanism should provide a defense against these attacks. But, in practice it is rarely possible to obtain a model with enough utility without increasing the privacy loss budget beyond the minimum needed to establish such guarantees. Instead, models are tested using empirical methods using simulated attacks to understand how much an adversary would be able to infer. Previous works on membership inference attacks only consider balanced priors, however, leading to a skewed understanding of inference risk in cases where models are likely to face adversaries with imbalanced priors. In this work, we develop a metric based on positive predictive value that captures the inference risk even in scenarios where the priors are skewed, and introduce a new attack strategy that shows models are vulnerable to inference attacks even in settings where previous attacks would be unable to infer anything useful.

Theoretical Contributions. Motivated by recent results [22, 28], we aim to develop more useful privacy metrics. Similarly to Liu et al. [28], we adopt a hypothesis testing perspective on differential privacy in which the adversary uses hypothesis testing on the differentially private mechanism's output to make inferences about its private training data. We use the recently proposed f-differential privacy notion (see Section 3.1) to bound the privacy leakage of the mechanism. Using this hypothesis testing framework, we tighten the theoretical bound on the advantage metric (Section 4.1). Then, we show that this metric alone does not suffice in most realistic scenarios since it does not consider the prior probability of the data distribution from which the adversary chooses records. We propose using positive predictive value (PPV) in conjunction with the advantage

***Corresponding Author: Bargav Jayaraman:** University of Virginia, USA, E-mail: bj4nq@virginia.edu
Lingxiao Wang: University of California Los Angeles, USA, E-mail: lingxw@ucla.edu
Katherine Knipmeyer: University of Virginia, USA, E-mail: kak9gsz@virginia.edu
Quanquan Gu: University of California Los Angeles, USA, E-mail: qgu@cs.ucla.edu
David Evans: University of Virginia, USA, E-mail: evans@virginia.edu

metric as it captures this notion, and provide a theoretical analysis of this metric (Section 4.2).

Empirical Contributions. We provide a threshold selection procedure that can be used to improve any threshold-based inference attack to better capture how an adversary with a particular goal would use the attack (Section 5.1). We use this procedure for the loss-based attack of Yeom et al. [46] and the confidence-based attack of Shokri et al. [38], as well as for two new attacks. We propose a novel inference attack strategy that samples points around the candidate input to gauge if it is near a local minimum in the loss function (Section 5.2). The Merlin attack uses this strategy to decide if an input is a member based on a threshold on the ratio of samples where the loss value increases. Our Morgan attack (Section 5.3) combines this with thresholds on the per-instance loss value. Finally, we use these attacks to empirically evaluate the privacy leakage of neural networks trained both with and without differential privacy on four multi-class data sets considering balanced and imbalanced prior data distribution (Section 7). Our main empirical findings include:

- Non-private models are vulnerable to high-confidence membership inference attacks in both balanced and imbalanced prior settings.
- PPV changes with the prior and hence it is a more reliable metric in imbalanced prior settings.
- The Morgan attack achieves higher PPV than Merlin, which already outperforms previous attacks.
- Private models can be vulnerable to our attacks, but only when privacy loss budgets are well above the theoretical guarantees.

2 Related Work

While statistical membership inference attacks were demonstrated on genomic data in the late 2000s [19, 36], the first membership inference attacks against machine learning models were performed by Shokri et al. [38]. In these attacks, the attacker exploits the model confidence reflecting overfitting to infer membership. Shokri et al. [38] consider the balanced prior setting and evaluate the attack success with an accuracy metric. The attacker trains shadow models similar to the target model, and uses these shadow models to train a membership inference model. Yeom et al. [46] proposed a simpler, but usually more effective, attack based on per-instance loss and proposed using membership advantage metric for attack evaluation as it has theoretical interpretation with differential privacy.

Yeom et al.'s membership advantage metric is useful for balanced prior settings, but not representative of true privacy leakage in realistic scenarios (as we demonstrate in Section 4). Rahman et al. [34] evaluate differentially private mechanisms against membership inference attacks and use accuracy and F-score as privacy leakage metrics. But they do not specify the theoretical relationship between their privacy leakage metrics and the privacy loss budgets (i.e., how the metric would scale with increasing privacy loss budget) necessary to gain insight as to what privacy loss budgets are safe even in the worst case scenarios. Jayaraman and Evans [22] evaluate the private mechanisms against both membership inference and attribute inference attacks using the advantage privacy leakage metric of Yeom et al. [46]. All the above works consider a balanced prior data distribution probability and hence are not applicable to settings where the prior probability is skewed.

Liu et al. [28] theoretically evaluate differentially private mechanisms using a hypothesis testing framework using precision, recall and F-score metrics. They give a theoretical relationship connecting these metrics to the differential privacy parameters (ϵ and δ) and give some insights for choosing the parameter values based on the background knowledge of the adversary. Recently, Balle et al. [4] provided hypothesis testing framework for analysing the relaxed variants of differential privacy that use Rényi divergence. However, neither of the above works provide empirical evaluation of privacy leakage of the private mechanisms. In another recent work, Farokhi and Kaafar [14] propose using conditional mutual information as the privacy leakage metric and derive its upper bound based on Kullback–Leibler divergence. Although they provide a relationship between this upper bound and the privacy loss budget, they do not evaluate the empirical privacy leakage in terms of the proposed metric. We provide a theoretical analysis of privacy leakage metrics and perform membership inference attacks under the more realistic assumptions of different prior data distribution probabilities and an adversary that can adaptively pick inference thresholds based on specific attack goals.

3 Differential Privacy

Here, we provide background on the differential privacy notions we use. Table 1 summarizes the notations we use throughout. Dwork et al. [12] introduced a formal notion of privacy that provides a probabilistic information-theoretic security guarantee:

Definition 3.1 (Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ is (ϵ, δ)*-differentially private* if for any pair of neighbouring data sets S, S' that differ by one record, and any set of outputs O,

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

Thus, the ratio of output probabilities across neighbouring data sets is bounded by the ϵ and δ parameters. The intuition behind this definition is to make any pairs of neighbouring data sets indistinguishable to the adversary given the information released.

From a hypothesis testing perspective [4, 11, 25, 28, 44], the adversary can be viewed as performing the following hypothesis testing problem given the ouput of either $\mathcal{M}(S)$ or $\mathcal{M}(S')$:

$$H_0 : \text{the underlying data set is } S,$$
$$H_1 : \text{the underlying data set is } S'.$$

According to the definition of differential privacy, given the information released by the private algorithm $\mathcal{M}$, the hardness of this hypothesis testing problem for the adversary is measured by the worst-case likelihood ratio between the distributions of the outputs $\mathcal{M}(S)$ and $\mathcal{M}(S')$. Following Wasserman and Zhou [44], a more natural way to characterize the hardness of this hypothesis testing problem is its type I and type II errors and can be formulated in terms of finding a rejection rule ϕ which trades off between type I and type II errors in an optimal way. In other words, for a fixed type I error α, the adversary tries to find a rejection rule ϕ that minimizes the type II error β. More specifically, recalling the definition of trade-off function from Dong et al. [11]:

Definition 3.2 (Trade-off Function)**.** For any two probability distributions P and Q on the same space, the *trade-off function* $T(P, Q) : [0, 1] \to [0, 1]$ is:

$$T(P, Q)(\alpha) = \inf\{\beta_{\phi} : \alpha_{\phi} \leq \alpha\},$$

where the infimum is taken over all (measurable) rejection rules, and α_{ϕ} and β_{ϕ} are the type I and type II errors for the rejection rule ϕ.

This definition suggests that the larger the trade-off function is, the harder the hypothesis testing problem will be. It has been established in Dong et al. [11] that a function $f : [0, 1] \to [0, 1]$ is a trade-off function if and only if it is convex, continuous, non-increasing, and $f(x) \leq 1 - x$ for $x \in [0, 1]$. Thus, differential privacy can be reformulated as finding the trade-off function f that limits the adversary's hypothesis testing power, i.e., it maximizes the adversary's type II error for any given type I error.

Notation	Description
$\mathcal{D} = \mathcal{X} \times \mathcal{Y}$	Distribution of records with features sampled form $\mathcal{X}$ and labels sampled from $\mathcal{Y}$
$S \sim \mathcal{D}^n$	Data set S consisting of n records, sampled i.i.d. from distribution $\mathcal{D}$
$\mathbf{z} \sim S$	Record z is picked uniformly from data set S
$\mathbf{z} \sim \mathcal{D}$	Record z is chosen according to distribution $\mathcal{D}$
$\mathcal{M}_S$	Model obtained by using a learning algorithm $\mathcal{M}$ over data set S
$\mathcal{A}$	Membership inference adversary
p	Probability of sampling a record from train set
γ	Test-to-train set ratio, $(1 - p)/p$
ϵ	privacy loss budget of DP mechanism
δ	Failure probability of DP mechanism
α	False positive rate (FPR) of inference adversary
β	False negative rate of inference adversary
ϕ	Decision threshold of inference adversary; also called rejection rule in hypothesis testing

Table 1. Notation

Several differentially private machine learning algorithms [8, 21, 23, 47] have been proposed that consume a small privacy loss budget ($\epsilon < 1$) without sacrificing the model accuracy for convex learning methods. Recent advances in composition analysis of differential private mechanisms [1, 7, 13, 32] have made private deep learning [1, 5, 6, 18, 20] possible with acceptable model utility, but still requiring large privacy loss budgets to make the guarantees provided by differential privacy insufficient to provide meaningful privacy.

3.1 *f*-Differential Privacy

The hypothesis testing formulation of differential privacy described above leads to the notion of f-differential privacy [11] (f-DP) which aims to find the optimal trade-off between type I and type II errors and will be used to derive the theoretical upper bounds of our proposed metrics for the privacy leakage.

Definition 3.3 (f-Differential Privacy)**.** Let f be a trade-off function. A mechanism $\mathcal{M}$ is f*-differentially private* if for all neighbouring data sets S and S':

$$T(\mathcal{M}(S), \mathcal{M}(S')) \geq f.$$

Note that in the above definition, we abuse the notations of $\mathcal{M}(S)$ and $\mathcal{M}(S')$ to represent their corresponding distributions. For an (ϵ, δ)-differentially private algorithm, the trade-off function $f_{\epsilon,\delta}$ is given by Lemma 3.1, which has been proved by Wasserman and Zhou [44] and Kairouz et al. [25]:

Lemma 3.1 ([25, 44])**.** *Suppose $\mathcal{M}$ is an (ϵ, δ)-differentially private algorithm, then for a false positive rate of α, the trade-off function is given by:*

$$f_{\epsilon,\delta}(\alpha) = \max\{0, 1 - \delta - e^{\epsilon}\alpha, e^{-\epsilon}(1 - \delta - \alpha)\}.$$

This lemma suggests that higher values of $f_{\epsilon,\delta}(\alpha)$ correspond to more privacy and perfect privacy would require $f_{\epsilon,\delta}(\alpha) = 1 - \alpha$. In addition, increasing ϵ and δ decreases $f_{\epsilon,\delta}(\alpha)$, reflecting the expected reduction in privacy.

3.2 Gaussian Differential Privacy

The Gaussian mechanism is a fundamental approach for achieving differential privacy, especially for differentially private deep learning [1]. More specifically, noisy stochastic gradient descent (SGD) and noisy Adam [2], i.e., adding Gaussian noise (Gaussian mechanism) to SGD and Adam, are often used as the underlying private optimizers for training neural networks with privacy guarantees. Therefore, precisely characterizing the privacy loss of the composition of Gaussian mechanisms and deriving its sub-sampling amplification results are of great interest. This motivates the notion of Gaussian differential privacy [11], which belongs to the family of f-DP with a single parameter μ that defines the mean of the Gaussian distribution.

Definition 3.4 (μ-Gaussian Differential Privacy)**.** A mechanism $\mathcal{M}$ is *μ-Gaussian differentially private* if for all neighbouring data sets S and S':

$$T(\mathcal{M}(S), \mathcal{M}(S')) \geq G_{\mu},$$

where $G_{\mu} = T(\mathcal{N}(0, 1), \mathcal{N}(\mu, 1))$.

In this definition, G_{μ} is a trade-off function and hence μ-GDP is identical to f-DP where $f = G_{\mu}$. Lemma 3.2, which is established in Dong et al. [11], gives the equation for computing G_{μ}:

Lemma 3.2. *Given that $\mathcal{M}$ is a μ-Gaussian differentially private algorithm, then for a false positive rate of α, the trade-off function is given as:*

$$G_{\mu}(\alpha) = \Phi(\Phi^{-1}(1 - \alpha) - \mu),$$

where Φ is the cumulative distribution function of standard normal distribution.

In our experiments, we use Gaussian differential privacy for training differentially private neural networks.

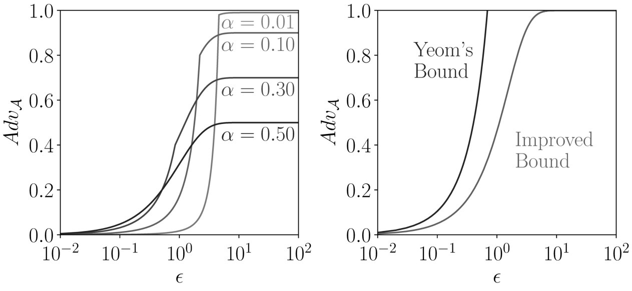
4 Measuring Privacy Leakage

To evaluate privacy leakage, we define an adversarial game inspired by Yeom et al.'s [46]. Unlike their game which assumes a balanced prior, our game factors in the prior membership distribution probability. The adversarial game models the scenario where an adversary has access to a model, $\mathcal{M}_S$, trained over a data set S, knowledge of the training procedure and data distribution, and wishes to infer whether a given input is a member of that training set.

Experiment 4.1 (Membership Experiment)**.** Assume a membership adversary, $\mathcal{A}$, who has information about the training data set size n, the distribution $\mathcal{D}$ from which the data set is sampled, and the prior membership probability p. The adversary runs this experiment:

1. Sample a training set $S \sim \mathcal{D}^n$ and train a model $\mathcal{M}_S$ over the training set S.
2. Randomly sample $b \in \{0, 1\}$, such that $b = 1$ with probability p.
3. If $b = 1$, then sample $\mathbf{z} \sim S$; else sample $\mathbf{z} \sim \mathcal{D}$.
4. Output 1 if $\mathcal{A}(\mathbf{z}, \mathcal{M}_S, n, \mathcal{D}) = b$; otherwise output 0.

Note that our experiment incorporates the prior probability p of sampling a record, compared to the setting of Yeom et al. that assumes balanced prior probability ($p = 0.5$). We consider skewed prior p as inferring membership is more important than inferring non-membership in our problem setting. This is different from the semantic security analogue where all messages are treated equally regardless of the skewness of the message distribution. For most practical scenarios (that is, where being exposed as a member carries meaningful risk to an individual), p is much smaller than 0.5. For instance, for a scenario of an epidemic outbreak, the training set could be the list of patients with the disease symptoms admitted at a hospital. The non-members can be the remaining population of the city or a district. Hence, assuming a balanced prior of $p = 0.5$ is not a realistic assumption, and it is important to develop a privacy metric that can be used to evaluate scenarios with lower (or higher) priors.



(a) $Adv_{\mathcal{A}}$ with varying α **(b)** Maximum $Adv_{\mathcal{A}}$ bound

Fig. 1. (a): $Adv_{\mathcal{A}}(\alpha)$ bounds for various privacy loss budgets ($\delta = 10^{-5}$). (b): Improving the bound on maximum advantage ($\delta = 0$). Improved bound uses Theorem 4.1 to get maximum advantage across all $0 < \alpha \leq 1$.

4.1 Membership Advantage

The membership advantage metric, *Adv*, was defined by Yeom et al. [46] as the difference between the true positive rate and the false positive rate for the membership inference adversary provided that $p = 0.5$ (i.e., balanced prior distribution). Yeom et al. showed that for an ϵ-differentially private mechanism, the theoretical upper bound for membership advantage is $e^{\epsilon} - 1$, which can be quite loose for higher ϵ values and is not defined for $e^{\epsilon} - 1 > 1$ since the metric proposed by Yeom et al. is only defined between 0 and 1. Moreover, the bound is not valid for (ϵ, δ)-differentially private algorithms which are more commonly used for private deep learning.

We derive a tighter bound for the membership advantage metric that is applicable to (ϵ, δ)-differentially private algorithms based on the notion of f-DP:

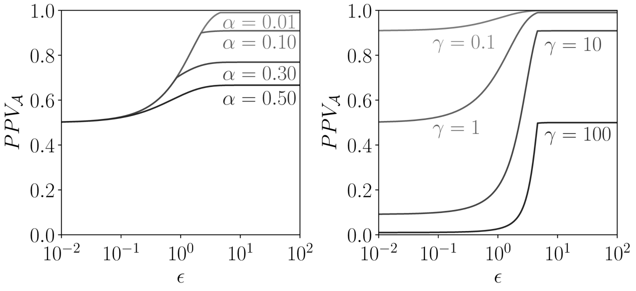
Theorem 4.1. *Let $\mathcal{M}$ be an (ϵ, δ)-differentially private algorithm. For any randomly chosen record* $\mathbf{z}$ *and fixed false positive rate α, the membership advantage of a membership inference adversary $\mathcal{A}$ is bounded by:*

$$Adv_{\mathcal{A}}(\alpha) \leq 1 - f_{\epsilon,\delta}(\alpha) - \alpha,$$

where $f_{\epsilon,\delta}(\alpha) = \max\left\{0, 1 - \delta - e^{\epsilon}\alpha, e^{-\epsilon}(1 - \delta - \alpha)\right\}$.

Proof of Theorem 4.1. The proof follows directly from Yeom at al.'s definition, $Adv_{\mathcal{A}}(\alpha) = TPR - FPR$, when we have balanced prior membership distribution, $p = 0.5$. For a given $FPR = \alpha$, we have $1 - TPR \geq f_{\epsilon,\delta}(\alpha)$ according to the definition of trade-off function (Definition 3.2 and Lemma 3.1). Therefore, $Adv_{\mathcal{A}}(\alpha) \leq 1 - f_{\epsilon,\delta}(\alpha) - \alpha$. □

Figure 1a shows the relationship between the false positive rate α of a given adversary and the upper bound of the advantage given by Theorem 4.1. This bound lies strictly between 0 and 1 and is tighter than the bound of Yeom et al. [46], as shown in Figure 1b. However, this metric is limited to balanced prior distribution of data and hence can overestimate (or underestimate) the privacy threat in any scenario where the prior probability is not 0.5. Thus, membership advantage alone is not a reliable way to measure the privacy leakage. Hence, we next propose the positive predictive value metric that considers the prior distribution of data.



(a) $PPV_{\mathcal{A}}$ with varying α **(b)** $PPV_{\mathcal{A}}$ with varying γ

Fig. 2. PPV bounds for various privacy loss budgets ($\delta = 10^{-5}$). For figure (a) $\gamma = 1$, and for figure (b) $\alpha = 0.01$.

4.2 Positive Predictive Value

Positive predictive value (PPV) gives the ratio of true members predicted among all the positive membership predictions made by an adversary (the precision of the adversary). For an (ϵ, δ)-differentially private algorithm, the PPV is bounded by the following theorem:

Theorem 4.2. *Let $\mathcal{M}$ be an (ϵ, δ)-differentially private algorithm and $\mathcal{A}$ be a membership inference adversary. For any randomly chosen record* $\mathbf{z}$ *and a fixed false positive rate of α, the positive predictive value of $\mathcal{A}$ is bounded by*

$$PPV_{\mathcal{A}}(\alpha, \gamma) \leq \frac{1 - f_{\epsilon,\delta}(\alpha)}{1 - f_{\epsilon,\delta}(\alpha) + \gamma\alpha},$$

where $f_{\epsilon,\delta}(\alpha) = \max\left\{0, 1 - \delta - e^{\epsilon}\alpha, e^{-\epsilon}(1 - \delta - \alpha)\right\}$, $\gamma = (1 - p)/p$, *and p is the prior membership probability defined in Membership Experiment 4.1.*

Proof of Theorem 4.2. According to the trade-off function definition (Definition 3.2 and Lemma 3.1), for a given $FPR = \alpha$, we have $1 - TPR \geq f_{\epsilon,\delta}(\alpha)$. Since $PPV_{\mathcal{A}}(\alpha, \gamma) = TP/(TP + FP)$, we can obtain:

$$PPV_{\mathcal{A}}(\alpha, \gamma) = \frac{TPR}{TPR + \gamma \cdot FPR} \leq \frac{1 - f_{\epsilon,\delta}(\alpha)}{1 - f_{\epsilon,\delta}(\alpha) + \gamma\alpha}.$$

□

Like membership advantage, the PPV metric is strictly bounded between 0 and 1. Moreover, the bound on PPV metric considers the prior distribution via γ, which gives the ratio of probability of selecting a non-member to a member. This allows the PPV metric to better capture the privacy threat across different settings. Figure 2a shows the effect of varying the false positive rate α and Figure 2b shows the effect of varying the prior distribution probability γ on the PPV metric. For example, for $\epsilon = 5, \delta = 10^{-5}, \alpha = 0.01, \gamma = 100$, the advantage metric can be as high as 0.98, while the PPV metric is close to 0.5 (i.e., coin toss probability). Thus, in such cases, advantage grossly overestimates the privacy threat.

5 Inference Attacks

While the previous section covers the metrics to evaluate privacy leakage, here we discuss about the membership inference attack procedures. In Section 5.1, we describe our threshold selection procedure for threshold-based inference attacks. Section 5.2 presents our threshold-based inference attack that perturbs a query record and uses the direction of change in per-instance loss of the record for membership inference. Section 5.3 presents our second attack that combines our first attack with the threshold-based attack of Yeom et al. [46].

5.1 Setting the Decision Threshold

The membership inference attacks we consider need to output a Boolean result for each test, converting a real number measure from a test into a Boolean that indicates whether or not a given input is considered a member. The effectiveness of an attack depends critically on the value of this decision threshold.

We introduce a simple procedure to select the decision threshold for any threshold-based attack where the adversary's goal is to maximize leakage for a given expected maximum false positive rate:

Procedure 5.1 (Finding the Decision Threshold). Given an adversary, $\mathcal{A}$, that knows information about a target model including the training data distribution $\mathcal{D}$, training set size n, training procedure, and model architecture, as well as knowing the prior distribution probability p for the suspected membership set, this procedure finds a threshold ϕ that maximizes the privacy leakage of the sampled data points for a given maximum false positive rate α.

1. Sample a training data set $\bar{S} \sim \mathcal{D}^n$ for training a model $\mathcal{M}_{\bar{S}}$.
2. Randomly sample $b \in \{0, 1\}$, such that $b = 1$ with probability p.
3. Sample record $\mathbf{z} \sim \bar{S}$ if $b = 1$, otherwise $\mathbf{z} \sim \mathcal{D}$.
4. Output the decision threshold, ϕ, that maximizes its true positive rate constrained to a maximum false positive rate of α for the inference attack, $\mathcal{A}(\mathbf{z}, \mathcal{M}_{\bar{S}}, n, \mathcal{D}, \phi)$.

Note that in comparison to Experiment 4.1, the adversary $\mathcal{A}$ takes an additional parameter ϕ, which is used to query the target model $\mathcal{M}_S$ to perform membership inference. Procedure 5.1 works for any threshold-based inference attack where an adversary knows the data distribution and model training process well enough to train its own models similar to the target model.

Application to Yeom's Attack. The membership inference attack of Yeom et al. [46] uses per-instance loss information for membership inference. Given a loss $\ell(\mathbf{z}, \mathcal{M}_S)$ on the query record $\mathbf{z}$, their approach classifies it as a member if the loss is less than the expected training loss. Using Procedure 5.1, we instead find a threshold ϕ for membership inference that corresponds to an expected maximum false positive rate α. In other words, if the per-instance loss $\ell(\mathbf{z}, \mathcal{M}_S) \leq \phi$, then $\mathbf{z}$ is classified as a member of the target model's training set S, otherwise it is classified as a non-member. We refer to this membership inference adversary as Yeom.

Application to Shokri's Attack. In the membership inference attack of Shokri et al. [38], the attacker first trains multiple shadow models similar to the target model, and then uses these shadow models to train an inference model for binary classification. We modify this attack by taking the softmax output of the inference model that indicates the model's prediction confidence, and use our threshold selection procedure on the model confidence. By default, the model predicts the input is a member if the confidence is above 0.5, which is equivalent to Shokri et al.'s original version. We vary this threshold between 0 and 1 according to Procedure 5.1, and refer to this inference adversary as Shokri.

5.2 Merlin

Procedure 5.1 can be used on any threshold-based inference attack. Here, we introduce a new threshold-based

Algorithm 1: Inference Using Direction of Change in Per-Instance Loss

```
$\mathcal{A}(\mathbf{z}, \mathcal{M}_S, n, \mathcal{D}, \phi)$:
  Input  : $\mathbf{z}$: input record, $\mathcal{M}_S$: model trained
           on data set $S$ of size $n$, $\mathcal{D}$: data
           distribution, $\phi$: decision function, $T$:
           number of repeat, $\sigma$: standard
           deviation parameter
  Output: membership prediction of $\mathbf{z}$ (0 or 1)
count $\leftarrow 0$ ;
for $T$ runs do
    $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ ; // Sample Gaussian noise
    if $\ell(\mathbf{z} + \boldsymbol{\xi}, \mathcal{M}_S) > \ell(\mathbf{z}, \mathcal{M}_S)$ then
        count $\leftarrow$ count $+ 1$ ;
    end
end
return $count/T \geq \phi$ ;          // 1 if 'member'
```

membership inference attack called Merlin[1] that uses a different approach to infer membership. This method checks if the per-instance loss of the record increases when perturbed with a small amount of noise. The intuition here is that due to overfitting, the target model's loss on a training set record will tend to be close to a local minimum, so the loss at perturbed points near the original input will be higher. For a non-member record, the loss is equally likely to either increase or decrease.

Algorithm 1 describes the attack procedure. For a query record $\mathbf{z}$, random Gaussian noise with zero mean and standard deviation σ is added and the change of loss direction is recorded. This step is repeated T times and the *count* is incremented each time the per-instance loss of the perturbed record increases. Though we use Gaussian noise, the algorithm works for other noise distributions as well. We also tried uniform distribution and observed similar results, but with different σ values. Both the parameters T and σ can be pre-tuned on a hold-out set to maximize the attacker's distinguishing power and fixed for the entire attack process. In our experiments, we find $T = 100$ and $\sigma = 0.01$ work well across all data sets. Finally, the query record $\mathbf{z}$ is classified as a member when $count/T \geq \phi$, where ϕ is a threshold that could be set by Procedure 5.1.

Comparison with Related Attacks. Although the intuition behind the Merlin is new, it has similarities with previous attacks that also involve sampling. Frederikson et al. [15] proposed a white-box attack for model inversion problem, which is different from the membership inference problem we consider, where the attacker has *count* information of all training instances and uses it to guess the most probable value for the sensitive attribute of the query training instance. This 'count' is different from the count used in Merlin attack. Long et al. [30] proposed a black-box model inversion attack that is similar to Merlin. While the Merlin attack considers the target point's environment in the input space, the attacks in Long et al. [30] consider the target point's environment in the logit-space, i.e., the output of the target network before the softmax is applied. As the logit-space is much more dense than the input space, Merlin is much more fine-grained, enabling it to detect membership where the logit-space attacks would not. Choo et al. [9] recently proposed a label-only membership inference attack which is similar to Merlin in the sense that they also use the model's behavior on *neighboring points* as part of a membership inference attack. The key difference is that they assume the neighboring points, which in their case are data augmentations of the target record, are also present in the training set, while we do not have any such assumptions for Merlin.

5.3 Morgan

Both Yeom and Merlin use different information for membership inference and hence do not necessarily identify the same member records. Some members are more vulnerable to one attack than the other, and different inputs produce false positives for each attack. Our observations of the distribution of the values from the Yeom and Merlin attacks (see Figure 6) motivate combining the attacks in a way that can maximize PPV by excluding points with very low per-instance loss. The intuition is that if the per-instance loss is extremely low, the Merlin attack will suggest a local minimum, but in fact it is a near-global minimum, which is not as strongly correlated with being a member. Hence, we introduce a combination of the Yeom and Merlin attacks, called Morgan[2], that combines both attacks to identify inputs that are most likely to be members.

The Morgan attack uses three thresholds: a lower threshold on per-instance loss ϕ_L, an upper threshold on per-instance loss ϕ_U, and a threshold on the ratio as used by Merlin, ϕ_M. Morgan classifies a record as mem-

[1] Backronym for **ME**asuring **R**elative **L**oss **I**n **N**eighborhood.

[2] **M**easuring l**O**ss, **R**elatively **G**reater **A**round **N**eighborhood.

ber if the per-instance loss of the record is between ϕ_L and ϕ_U, both inclusive, and has a Merlin ratio of at least ϕ_M. The ϕ_U and ϕ_M thresholds are set using the standard threshold selection procedure for the Yeom and Merlin attacks respectively, by varying their α values. A value for ϕ_L is found using a grid search to find the maximum PPV possible in conjunction with ϕ_U and ϕ_M thresholds, and selecting the lowest value for ϕ_L that achieves that PPV to maximize the number of members identified. Note that all three thresholds are selected together to maximize the PPV on a separate holdout set that is disjoint from the target training set, as is done in our threshold selection procedure 5.1. As reported in Table 2, this exposes some members with 100% PPV for both RCV1X and CIFAR-100. Section 7 reports on Morgan's success on identifying the most vulnerable records with > 95% PPV at balanced prior and with > 90% PPV in skewed prior cases ($\gamma > 1$).

6 Experimental Setup

This section describes the data sets and models used, along with the training procedure. We evaluate our methods on both standard (non-private) models and models trained using differential privacy mechanisms. We focus on differentially private models since our theoretical bounds apply to these models. Although several other defenses have been proposed, such as dropout, model stacking or MemGuard [24], our theoretical bounds do not apply to them and we do not include them in our evaluation.[3]

Table 2 summarizes the data sets used and the performance of non-private models trained over each data set, and the leakage from the most effective membership inference attack (Morgan). In the balanced prior setting ($\gamma = 1$), some members are exposed with very high confidence (>95% PPV) for all the test data sets. The membership inference is significant even in the imbalanced prior case, when $\gamma = 10$. We defer discussion of these results to Section 7.

Data Sets. Multi-class classification tasks are more vulnerable to membership inference, as shown in prior works on both black-box [38, 46] and white-box [33] attacks. Hence, we select four multi-class classification tasks for our experiments. Although these data sets are public, they are representative of data sets that contain potentially sensitive information about individuals.

– Purchase-100X: Shokri et al. [38] created Purchase-100 data set by extracting customer transactions from Kaggle's acquire valued customers challenge [10]. The authors *arbitrarily* selected 600 items from the transactions data and considered only those customers who purchased at least one of the 600 items. Their resulting data set consisted of 197,000 customer records with 600 binary features representing the customer purchase history. The records are clustered into 100 classes, each representing a unique purchase style, such that the goal is to predict a customer's purchase style. Since we needed more records for our experiments with the $\gamma = 10$ setting, we curated our own data set by following the same procedure but instead of 600 arbitrary items taking the 600 *most frequently* purchased items. This resulted in an expanded, but similar, data set with around 300,000 customer records which we call Purchase-100X.

– Texas-100: The Texas hospital data set, also used by Shokri et al. [38], consists of 67,000 patient records with 6,000 binary features where each feature represents a patient's medical attribute. This data set also has 100 output classes where the task is to identify the main procedure that was performed on the patient. This data set is too small for tests with high γ settings, but a useful benchmark for the other settings.

– RCV1X: The Reuters RCV1 corpus data set [27] is a collection of Reuters newswire articles with more than 800,000 documents, a 47,000-word vocabulary and 103 classes. The original 103 classes are arranged in a hierarchical manner, and each article can belong to more than one class. We follow data pre-processing procedures similar to Srivastava et al. [41] to obtain a data set such that each article only belongs to a single class. The final data set we use has 420,000 articles, 2,000 most frequent words represented by their term frequency–inverse document frequency (TFIDF) which are used as features and 52 classes. We call our expanded data set RCV1X.

– CIFAR-100: We use the standard CIFAR-100 [26] data set used in machine learning which consists of 60,000 images of 100 common world objects. The task is to identify an object based on the input RGB image consisting of 32 × 32 pixels. Although the privacy issue here is not clear, we include this data set in our experiments because it is used as a benchmark in many privacy works.

[3] Our attacks and experimental tests do, however, and it will be interesting to see how effective non-DP defenses are against our attacks, so we do plan to include evaluations of other defenses in future work.

Data set	#Features	#Classes	Train Acc	Test Acc	Leakage at Balanced Prior	Leakage at $\gamma = 10$ Prior
Purchase-100X	600	100	1.00	0.71	11 ± 3 (98.0 ± 4.0 PPV)	8 ± 6 (97.5 ± 5.0 PPV)
Texas-100	6,000	100	1.00	0.53	55 ± 25 (95.7 ± 4.6 PPV)	*(data set too small to test)*
RCV1X	2,000	52	1.00	0.84	41 ± 27 (100.0 ± 0.0 PPV)	8 ± 8 (93.0 ± 9.8 PPV)
CIFAR-100	3,072	100	0.48	0.18	2 ± 1 (100.0 ± 0.0 PPV)	*(data set too small to test)*

Table 2. Summary of data sets and results for non-private models. Leakage is the number of members identified out of 10,000 with near-certain confidence out of 10,000 members by Morgan while maximizing the PPV metric, averaged across five runs.

All the above data sets are pre-processed such that the ℓ_2 norm of each record is bounded by 1. This is a standard pre-processing procedure that improves model performance that is used by many prior works [8, 23].

Model Architecture. We train neural networks with two hidden layers using ReLU activation. Each hidden layer has 256 neurons and the output layer is a softmax layer. Several previous works used similar multilayer ReLU network architectures to analyze privacy-preserving machine learning [1, 37, 38]. Details on hyperparameters can be found in Appendix A. Table 2 includes the training and test accuracy of non-private models across the four data sets.[4] Although we tuned the model hyperparameters to maximize the test accuracy for each data set, there is a considerable gap between the training and test accuracy. This generalization gap indicates that the model overfits the training data, and hence, there is information in the model that could be exploited by inference attacks.

Private Model Training. We evaluate the model accuracy of private neural network models trained on different data sets. We vary the privacy loss budget ϵ between 0.1 and 100 for differentially private training and repeat the experiments five times for all the settings to report the average results.

We report the *accuracy loss*, which gives the relative loss in test accuracy of private models with respect to non-private baseline:

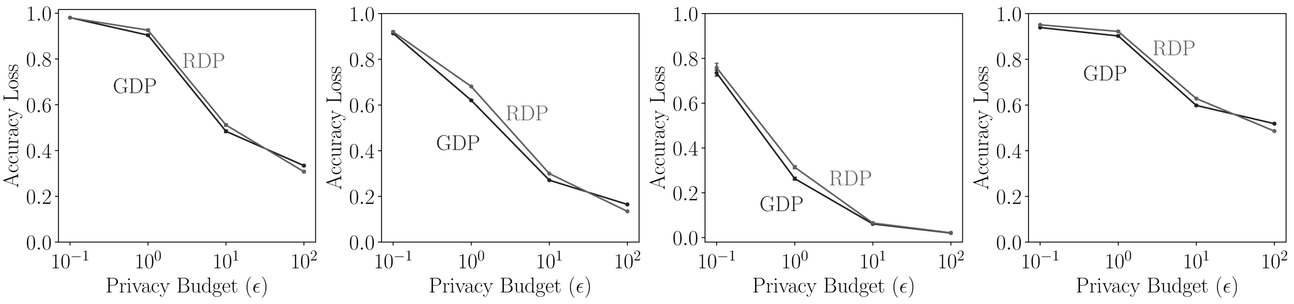
$$\textit{Accuracy Loss} = 1 - \frac{\textit{Accuracy of Private Model}}{\textit{Accuracy of Non-Private Model}}$$

Figure 3 gives the accuracy loss of differentially private models trained on different data sets with varying privacy loss budgets. The private models are trained using the gradient perturbation mechanism where the gradients at each epoch are clipped and Gaussian noise is added to preserve privacy. The privacy accounting for composition of mechanisms is done via both Gaussian differential privacy (GDP) [11] and the prior state-of-the-art Rényi differential privacy (RDP) [32]. As shown in the figure, the GDP mechanism has a lower accuracy loss for $\epsilon \leq 10$ due to its tighter privacy analysis. The GDP composition theorem requires that the individual mechanisms be highly private, and hence it is hard to reduce noise for $\epsilon > 10$ without increasing the failure probability δ. For all the data sets, GDP performs better than RDP, hence we only report the results for GDP in the remaining experiments.

7 Empirical Results

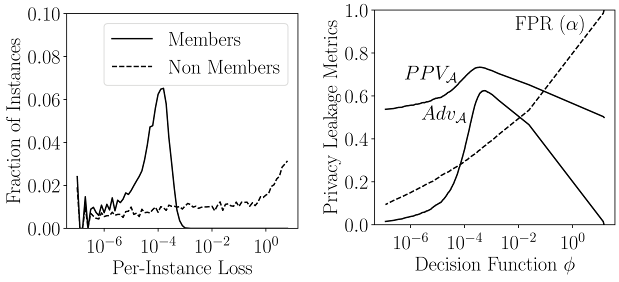
In this section, we evaluate our threshold selection procedure (Procedure 5.1) across the four inference attacks. We first consider the Yeom attack, and show that our threshold selection procedure can be used to obtain thresholds that achieve particular attacker goals, such as maximizing the PPV or membership advantage metric, or minimizing the false positive rate. Next, we use our threshold selection procedure on the Shokri attack and discuss the results in Section 7.2. In Section 7.3 we evaluate the Merlin attack using the same threshold selection procedure, and find that it achieves higher PPV metric compared to both Yeom and Shokri. Then, Section 7.4 shows how the Morgan attack achieves higher PPV by combining aspects of both Yeom and Merlin. Results in the first four subsections focus on non-private models and balanced prior scenarios. In Section 7.5 we evaluate the attacks on differentially private models. Section 7.6 presents results for scenarios with imbalanced priors. The results show that non-private models are vulnerable to our proposed attacks, especially Morgan, even in the skewed prior settings. Private models are vulnerable in the balanced prior setting if the privacy loss budget is set beyond theoretical guarantees.

[4] As with all of the experimental results we report in this paper, the results are averaged over five runs in which the target model is trained from the scratch for each run.



(a) Purchase-100X **(b)** Texas-100 **(c)** RCV1X **(d)** CIFAR-100

Fig. 3. Accuracy loss comparison of private models trained with different privacy analyses.



(a) Loss Distribution. **(b)** Yeom Performance.

Fig. 4. Analysis of Yeom on non-private model trained on Purchase-100X with balanced prior. The x-axis shows the per-instance loss on a logarithmic scale from 10^{-7} to 10^{1} where the buckets are in the range $(10^{-7}, 10^{-6.9})$, $(10^{-6.9}, 10^{-6.8})$, and so on up to $(10^{0.9}, 10^{1})$.

7.1 Yeom Attack

The Yeom attack uses a fixed threshold on per-instance loss for its membership inference test. A query record is classified as a member if its per-instance loss is less than the selected threshold. We show that the adversary can achieve better privacy leakage, specific to particular attack goals, by using our threshold selection procedure.

Results on Purchase-100X. Figure 4a shows the distribution of per-instance loss of members and non-members for a non-private model trained on Purchase-100X. Per-instance losses of members are concentrated close to zero, and most of the loss values are less than 0.001. Whereas for non-members, the loss values are spread across the range. This suggests that a larger fraction of members will be identified by the attacker with high precision (PPV) for loss thresholds less than 0.001, and hence the privacy leakage will be high.

Another notable observation is that out of the 10,000 test records there are 959.2 ± 23.5 non-members (average across five runs) with zero loss, and hence the minimum achievable false positive rate is around 10%. This is reflected in Figure 4b, which shows the effect of selecting different loss thresholds on the privacy leakage metrics. An attacker can use our threshold selection procedure to choose a loss threshold to meet specific attack goals, such as minimizing the false positive rate (Min FPR), or achieving a fixed false positive rate (Fixed FPR), or maximizing either of the privacy leakage metrics (Max $PPV_{\mathcal{A}}$ and Max $Adv_{\mathcal{A}}$). Table 3 summarizes these scenarios and compares their thresholds with the threshold selected by the method of Yeom et al. (Fixed ϕ). For Fixed FPR, we consider an attacker with a false positive rate of 1% ($\alpha = 1\%$).

The attacker uses Procedure 5.1 to find the loss threshold, ϕ, corresponding to $\alpha = 1\%$, which it uses for membership inference on the target set. However, since the minimum achievable false positive rate for Yeom on Purchase-100X is 10%, this attack fails to find a suitable threshold. For maximizing PPV or advantage, the attacker can use the threshold selection procedure with varying α values and choose the threshold ϕ that maximizes the required privacy metric. In comparison, Fixed ϕ uses expected training loss as threshold which does not necessarily maximize the privacy leakage. As the results in the table demonstrate, an attacker can accomplish different attack goals, and achieve increased privacy leakage, using the Yeom attack with thresholds chosen using our threshold selection procedure.

Results on Other Data Sets. Table 4 compares the performance of Yeom against non-private models across the Texas-100, RCV1X and CIFAR-100 data sets. We observe similar trends of privacy leakage corresponding to the selected thresholds for these data sets as we did for Purchase-100X so present most of the results for these data sets in Appendix B, and only discuss some notable differences here.

For Texas-100, Yeom can achieve false positive rates as low as 3%. The attack performance on this data set is

		α	ϕ	**Actual FPR**	**Actual TPR**	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$
Yeom	**Fixed FPR**	1.00	-	-	-	-	-
	Min FPR	10.00	0	5.7 ± 4.7	6.7 ± 5.5	1.0 ± 0.8	32.5 ± 26.5
	Fixed ϕ	-	$(1.0 \pm 0.0) \times 10^{-4}$	29.9 ± 0.3	63.7 ± 0.2	33.8 ± 0.3	68.1 ± 0.2
	Max $PPV_{\mathcal{A}}$	35.00	$(3.7 \pm 0.3) \times 10^{-4}$	35.3 ± 0.5	95.5 ± 1.1	60.2 ± 0.8	73.0 ± 0.2
	Max $Adv_{\mathcal{A}}$	37.00	$(6.0 \pm 0.4) \times 10^{-4}$	37.3 ± 0.5	99.2 ± 0.2	61.9 ± 0.3	72.7 ± 0.2
Yeom CBT	**Min FPR**	0.01	$0, 0, 6.7 \times 10^{-6}$	0.1 ± 0.0	0.3 ± 0.1	0.2 ± 0.1	73.4 ± 5.0
	Max $PPV_{\mathcal{A}}$	0.01	$0, 0, 6.7 \times 10^{-6}$	0.1 ± 0.0	0.3 ± 0.1	0.2 ± 0.1	73.4 ± 5.0
	Fixed FPR	1.00	$0, 0, 6.7 \times 10^{-6}$	0.1 ± 0.0	0.3 ± 0.1	0.2 ± 0.1	73.4 ± 5.0
	Fixed ϕ	-	$(0.2, 0.9, 2.4) \times 10^{-4}$	29.7 ± 0.3	62.1 ± 1.8	32.4 ± 1.6	67.6 ± 0.5
	Max $Adv_{\mathcal{A}}$	55.00	$(1.1, 4.6, 18.7) \times 10^{-4}$	36.9 ± 0.2	98.4 ± 0.3	61.6 ± 0.5	72.7 ± 0.2
Shokri	**Min FPR**	0.02	1.06 ± 0.38	0.1 ± 0.1	0.1 ± 0.1	0.0 ± 0.1	33.2 ± 31.7
	Fixed FPR	1.00	0.80 ± 0.02	1.2 ± 0.1	3.0 ± 0.4	1.8 ± 0.5	71.9 ± 4.2
	Max $PPV_{\mathcal{A}}$	1.92	0.78 ± 0.01	2.2 ± 0.1	6.0 ± 0.6	3.8 ± 0.5	73.4 ± 1.6
	Fixed ϕ	-	0.50 ± 0.00	48.6 ± 1.0	99.2 ± 0.8	50.6 ± 0.5	67.1 ± 0.3
	Max $Adv_{\mathcal{A}}$	47.30	0.50 ± 0.04	48.6 ± 0.4	99.2 ± 0.9	50.6 ± 0.7	67.1 ± 0.2
Shokri CBT	**Fixed FPR**	1.00	-	-	-	-	-
	Min FPR	2.00	0.5, 0.8, 1.8	0.4 ± 0.1	0.4 ± 0.2	0.1 ± 0.1	53.8 ± 5.0
	Max $PPV_{\mathcal{A}}$	40.00	0.4, 0.7, 1.1	36.6 ± 0.6	94.0 ± 0.9	57.5 ± 0.4	72.0 ± 0.2
	Max $Adv_{\mathcal{A}}$	50.00	0, 0.7, 1.1	39.0 ± 0.4	98.6 ± 0.8	59.6 ± 0.4	71.7 ± 0.1
	Fixed ϕ	-	0.5, 0.5, 0.5	48.6 ± 1.0	99.2 ± 0.8	50.6 ± 0.5	67.1 ± 0.3
Merlin	**Min FPR**	0.01	0.88 ± 0.01	0.0 ± 0.0	0.1 ± 0.0	0.1 ± 0.0	93.4 ± 6.3
	Max $PPV_{\mathcal{A}}$	0.01	0.88 ± 0.01	0.0 ± 0.0	0.1 ± 0.0	0.1 ± 0.0	93.4 ± 6.3
	Fixed FPR	1.00	0.78 ± 0.00	0.7 ± 0.1	4.2 ± 0.1	3.4 ± 0.2	85.0 ± 2.0
	Max $Adv_{\mathcal{A}}$	31.00	0.60 ± 0.00	30.5 ± 0.3	51.1 ± 0.1	20.6 ± 0.2	62.6 ± 0.2
Morgan	**Max $PPV_{\mathcal{A}}$**	-	3.4×10^{-5}, 6.0×10^{-4}, 0.88	0.0 ± 0.0	0.1 ± 0.0	0.1 ± 0.0	98.0 ± 4.0

Table 3. Thresholds selected against non-private models trained on Purchase-100X with balanced prior. The results are averaged over five runs such that the target model is trained from the scratch for each run. Yeom CBT uses class-based thresholds, where ϕ shows the triplet of minimum, median and maximum thresholds across all classes. All values, except ϕ, are in percentage.

comparable to that of Purchase-100X. For RCV1X, the attack success rate is substantially lower than that for the other data sets. This is because, unlike the other data sets which have 100 classes, RCV1X is a 52-class classification task. As reported in prior works [39, 46], success of membership inference attack is proportional to the complexity of classification task. We further note that the maximum PPV that can be achieved by Yeom on RCV1X is only around 58%, at which point the membership advantage is close to 27%. This gives credence to our claim that membership advantage should not be solely relied on as a measure of inference risk. While membership advantage can be high, the privacy leakage is negligible for balanced priors when the PPV is close to 50%. Later in Section 7.6 we show that this phenomenon is prevalent across all data sets when the prior is imbalanced.

Yeom's performance on CIFAR-100 is similar to that on Purchase-100X and Texas-100 data sets. Since the model does not completely overfit on CIFAR-100, the distribution of loss values for both members and non-members are not far apart, and as a consequence Yeom is able to achieve much lower false positive rates.

Using Class-Based Thresholds. Recently, Song and Mittal [40] demonstrated that the approach of Yeom et al. [46] can be further improved by using class-based thresholds instead of one global threshold on loss values. We implement this approach, using our threshold setting algorithm to independently set the threshold for each class (referred as Yeom CBT). This enables finding class-based thresholds corresponding to smaller α values, as seen for the minimum FPR ($\alpha = 0.01$) and fixed FPR ($\alpha = 1$) cases for Purchase-100X in Table 3. Nonetheless, the maximum PPV still does not increase much beyond Yeom on Purchase-100X, with the largest increase being from 73.0% to 73.4%. For other data sets, though, this technique improves the maximum PPV of Yeom. For Texas-100, the PPV increases from 76% to 92%, for RCV1X, the PPV increases from 58% to 93% and for CIFAR-100, the PPV increases from 73% to 81% (see Appendix B). However, the maximum PPV never exceeds beyond Merlin or Morgan. While Song and Mittal [40] also showed the application of their class-based thresholds on other metrics such as model confidence and modified entropy, their experimental results show that these approaches achieve similar attack per-

		Texas-100			RCV1X			CIFAR-100		
		α	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$	α	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$	α	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$
Yeom	Fixed FPR	1.00	-	-	1.00	-	-	1.00	0.7 ± 0.3	65.1 ± 2.6
	Min FPR	3.00	0.4 ± 0.9	12.0 ± 24.0	33.00	0.4 ± 0.8	10.3 ± 20.5	0.01	0.0 ± 0.0	33.3 ± 27.9
	Fixed ϕ	-	51.3 ± 2.6	75.0 ± 1.6	-	26.9 ± 2.7	58.0 ± 0.8	-	33.0 ± 1.6	70.3 ± 1.1
	Max $PPV_{\mathcal{A}}$	26.00	59.2 ± 11.7	76.1 ± 1.6	67.00	24.8 ± 5.0	57.9 ± 1.0	12.00	19.0 ± 1.6	72.7 ± 0.8
	Max $Adv_{\mathcal{A}}$	31.00	62.9 ± 7.7	75.0 ± 0.6	70.00	25.1 ± 3.2	57.7 ± 0.6	39.00	37.2 ± 1.8	66.5 ± 0.6
Shokri	Min FPR	0.01	1.0 ± 0.5	72.6 ± 8.6	0.01	0.6 ± 0.2	91.7 ± 4.2	0.01	16.5 ± 1.9	64.9 ± 0.7
	Max $PPV_{\mathcal{A}}$	0.70	13.8 ± 1.1	89.4 ± 1.5	0.01	0.6 ± 0.2	91.7 ± 4.2	0.01	16.5 ± 1.9	64.9 ± 0.7
	Fixed FPR	1.00	16.0 ± 1.3	88.9 ± 1.7	1.00	4.6 ± 0.6	84.5 ± 1.8	1.00	24.6 ± 0.8	63.0 ± 0.4
	Fixed ϕ	-	64.0 ± 1.4	74.1 ± 1.3	-	24.0 ± 0.8	57.3 ± 0.4	-	26.0 ± 0.8	62.5 ± 0.4
	Max $Adv_{\mathcal{A}}$	31.00	64.1 ± 1.2	74.7 ± 0.9	75.00	24.2 ± 0.5	58.0 ± 0.4	8.00	26.9 ± 0.9	61.3 ± 0.4
Merlin	Min FPR	0.01	0.1 ± 0.1	51.9 ± 42.4	0.01	0.2 ± 0.0	98.8 ± 2.4	0.01	0.0 ± 0.0	51.4 ± 32.0
	Max $PPV_{\mathcal{A}}$	0.06	0.3 ± 0.2	92.0 ± 4.5	0.01	0.2 ± 0.0	98.8 ± 2.4	0.90	1.6 ± 0.5	75.0 ± 2.6
	Fixed FPR	1.00	4.9 ± 1.3	87.8 ± 2.7	1.00	2.6 ± 0.7	81.7 ± 4.3	1.00	1.8 ± 0.5	74.7 ± 1.7
	Max $Adv_{\mathcal{A}}$	36.00	37.8 ± 1.5	68.0 ± 0.8	26.00	11.6 ± 2.3	59.5 ± 2.0	39.00	27.7 ± 1.3	63.3 ± 0.3
Morgan	Max $PPV_{\mathcal{A}}$	-	0.5 ± 0.2	95.7 ± 4.6	-	0.4 ± 0.3	100.0 ± 0.0	-	0.0 ± 0.0	100.0 ± 0.0

Table 4. Comparing attacks on non-private models for balanced prior. All values are percentages ($\alpha = 0.01$ means 1 out of 10,000).

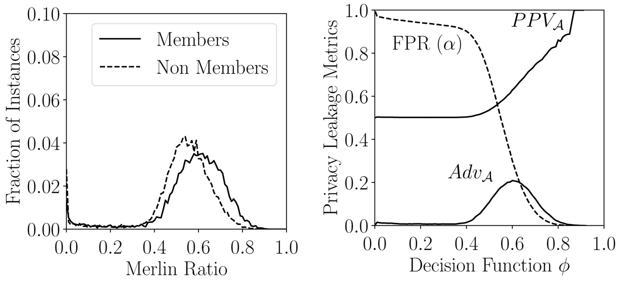
formance to the CBT on per-instance loss metric. Hence, we do not include the CBT results for other metrics.

7.2 Shokri Attack

The Shokri attack [38] requires training multiple shadow models on hold-out data sets similar to the target model. These shadow models are used to train an inference model that outputs a confidence value between 0 and 1 for membership inference, where 1 indicates member. We use the experimental setting of Jayaraman and Evans [22] to train five shadow models with the same architecture and hyperparameter settings of the target model. The inference model is a two-layer neural network with 64 neurons in each hidden layer. As with the Yeom attack, our threshold selection procedure can be used to increase privacy leakage for Shokri.

Results on Purchase-100X. Table 3 shows the privacy leakage of Shokri for different attack goals. The original attack of Shokri et al. (Fixed ϕ) uses a threshold of 0.5 on the inference model confidence and achieves close to 50% membership advantage, but has a PPV of around 67%. Using our threshold setting procedure to maximize PPV, Shokri achieves PPV of over 73%, which is comparable to the Yeom attack.

Results on Other Data Sets. Table 4 shows the results of Shokri across multiple data sets. The Shokri attack performance varies considerably across different data sets when compared to the Yeom attack. While Shokri achieves higher PPV than Yeom on Texas-100 and RCV1X, reflecting significant privacy risk on these data sets, Yeom outperforms Shokri on CIFAR-100. However, Merlin and Morgan consistently achieve higher PPV than both Yeom and Shokri (see Sections 7.3 and 7.4).



(a) Merlin Ratio Distribution. (b) Merlin Performance.

Fig. 5. Analysis of Merlin on non-private model trained on Purchase-100X with balanced prior.

Using Class-Based Thresholds. We also use class-based thresholds for Shokri attack and include the results for Purchase-100X in Table 3 (called Shokri CBT). However, we do not observe any significant improvement in privacy leakage over the Shokri attack. While the maximum membership advantage increases from 50% to around 60%, the maximum PPV is still close to 72%. We observe similar behaviour across other data sets.

7.3 Merlin Attack

Next, we perform inference attacks using the Merlin (Algorithm 1) where the attacker perturbs a record with random Gaussian noise of magnitude $\sigma = 0.01$ and notes the direction of change in loss. This process is repeated

$T = 100$ times and the attacker counts the number of times the loss increases out of T trials to find the Merlin ratio, $count/T$. If the Merlin ratio exceeds a threshold, then the record is classified as a member. As with the Yeom and Shokri experiments, we use Procedure 5.1 to select a suitable threshold.

Results on Purchase-100X. Figure 5a shows the distribution of Merlin ratio for member and non-member records for a non-private model trained on the Purchase-100X data set. The average Merlin ratio is 0.57 ± 0.17 for member records, whereas for the non-member records it is 0.52 ± 0.16. A peculiar observation is that the Merlin ratio is zero for a considerable fraction of members and non-members. For these non-member records, the loss is very high to begin with and hence it never increases for the nearby noise points. Whereas for the member records, the loss value does not change even with addition of noise. As mentioned in step 5 of Algorithm 1, we only check if the loss increases upon perturbation since we believe that equality is not a strong indicator of membership. Hence these outliers indicate regions where the loss doesn't change, not points where it always decreases.

Figure 5b shows the attack performance with varying thresholds. Merlin can achieve much higher PPV than Yeom and Shokri. Table 3 summarizes the thresholds selected by Merlin with different attack goals and compares the performance with Yeom and Shokri. While Yeom can only achieve a minimum false positive rate of 10% on this data set, Merlin can achieve false positive rate as low as 0.01%. Thus Merlin is successful at a fixed false positive rate of 1% where Yeom fails. Another notable observation is that Merlin can achieve close to 93% PPV, while the maximum possible PPV achievable via Yeom and Shokri (including their CBT versions) is under 74%. Thus, this attack is more suitable for scenarios where attack precision is preferred.

Results on Other Data Sets. Table 4 compares the membership inference attack performance against non-private models across the other data sets. The Merlin attack consistently achieves higher PPV than Yeom and Shokri across all the data sets. Merlin is more successful on Texas-100 compared to Purchase-100X, as the gap between Merlin ratio distribution of member records and non-member records is high for Texas-100 (see Appendix B for more analysis). More surprisingly, while Yeom is less successful on RCV1X, we find that Merlin still manages to achieve a very high PPV that even exceeds the PPV of Shokri (see Table 4). Thus, Merlin poses a credible privacy threat even in scenarios where Yeom fails. However, Merlin does not perform significantly better than Yeom and Shokri on CIFAR-100 since the per-instance loss of members is high on this data set and hence the members are not at local minimum. Appendix B provides more details on all these results.

Using Class-Based Thresholds. We also tried class-based thresholds for Merlin, like we did for Yeom and Shokri. However, we found that this approach does not benefit Merlin as the individual classes do not have enough records to provide meaningful thresholds. Using class-based thresholds for Merlin increases the advantage metric from 0.1% to 2.8%, but decreases the maximum achievable PPV from around 93.4% to 83.1%. We observed similar behavior across different thresholds.

7.4 Morgan Attack

The Morgan attack (Section 5.3) combines both Yeom and Merlin attacks to identify the most vulnerable members. Recall that Morgan classifies a record as member if its per-instance loss is between ϕ_L and ϕ_U and if the Merlin ratio is at least ϕ_M.

Results on Purchase-100X. Figure 6a shows the loss and Merlin ratio for members and non-members for one run of non-private model training in balanced prior. As shown, a fraction of members are clustered between 3.4×10^{-5} and 6.0×10^{-4} loss and with Merlin ratio at least 0.88, and in this region there are very few non-members. Thus, Morgan can target these vulnerable members whereas Yeom and Merlin fail to do, being restricted to a single threshold. As reported in Table 3, Morgan succeeds at achieving around 98% PPV while Yeom and Shokri only achieve 73% PPV at maximum on Purchase-100X whereas Merlin achieves 93% PPV.

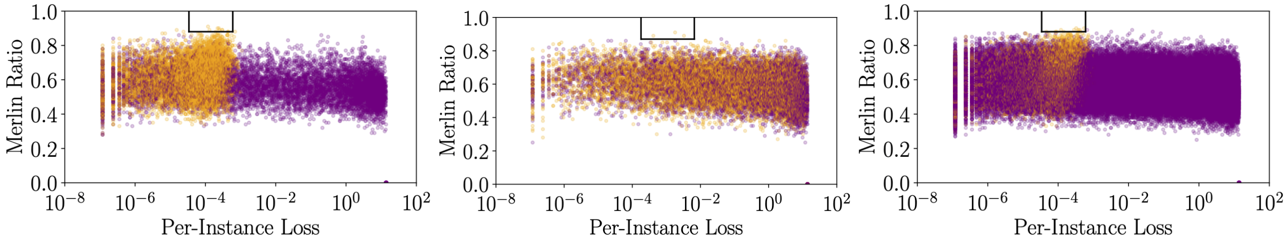
Results on Other Data Sets. Morgan exposes members with 100% PPV in our experiments against non-private models for the RCV1X and CIFAR-100 datasets, and exceeds 95% PPV for Texas-100 (Table 4). Morgan benefits by using multiple thresholds and is able to identify the most vulnerable members with close to 100% confidence. Further discussion on these results can be found in Appendix B.

7.5 Impact of Privacy Noise

So far, all results we have reported are for inference attacks on models trained without any privacy protections. We also evaluated membership inference attacks

	Yeom			Merlin			Morgan	
ϵ	α	ϕ	Max $PPV_{\mathcal{A}}$	α	ϕ	Max $PPV_{\mathcal{A}}$	ϕ	Max $PPV_{\mathcal{A}}$
1	0.03	$(9.6 \pm 1.1) \times 10^{-2}$	71.4 ± 25.7	0.12	0.87	67.2 ± 12.8	2.1, 4.3, 0.87	71.4 ± 17.2
10	0.90	$(3.7 \pm 1.5) \times 10^{-5}$	59.4 ± 2.4	0.02	0.88	79.7 ± 17.9	4.5×10^{-5}, 0.011, 0.88	95.0 ± 10.0
100	24.00	$(1.1 \pm 0.2) \times 10^{-2}$	60.5 ± 0.2	0.03	0.88	80.3 ± 24.8	1.8×10^{-4}, 0.0066, 0.87	93.3 ± 13.3

Table 5. Attacks against private models (Purchase-100X, balanced prior). α and PPV values are percentages. Standard deviation is not shown for Merlin's ϕ: across the five runs, it does not change for $\epsilon = 1$, but changes by ±0.01 for $\epsilon = 10$ and $\epsilon = 100$.



(a) Non-private model at $\gamma = 1$ **(b)** Private model at $\gamma = 1, \epsilon = 100$ **(c)** Non-private model at $\gamma = 10$

Fig. 6. Comparing loss and Merlin ratio side-by-side on Purchase-100X. Members and non-members are denoted by orange and purple points respectively. The boxes show the thresholds found by the threshold selection process (without access to the training data, but with the same data distribution), and illustrate the regions where members are identified by Morgan with high confidence.

against the private models and found the models to be vulnerable to Merlin and Morgan at privacy loss budgets high enough to train useful models. Like the experiments with non-private models, here also we repeat the experiments five times and report average results and standard error. In each run, we train a private model from scratch and perform the attack procedure on it.

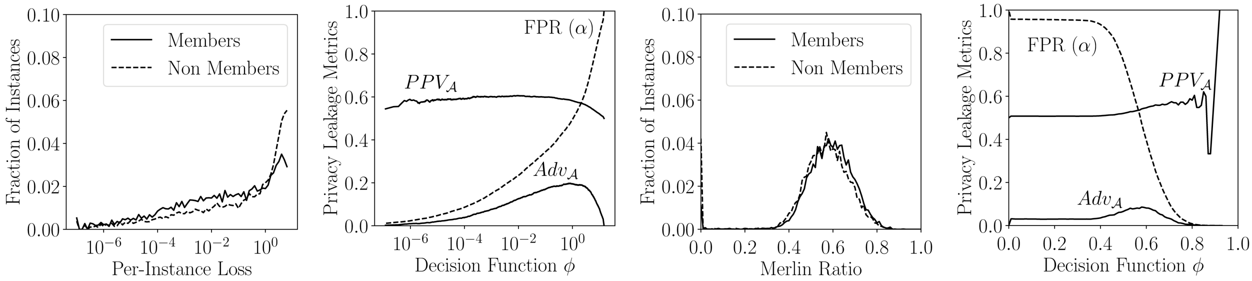
Table 5 compares the maximum PPV achieved by Yeom, Merlin and Morgan against private models trained on Purchase-100X with varying privacy loss budgets.[5] As expected, the privacy leakage increases with the privacy loss budget. Merlin and Morgan both achieve high PPV for privacy loss budgets, $\epsilon \geq 10$ (large enough to offer no meaningful privacy guarantee, but this is still smaller than needed to train useful models). Morgan has higher PPV on average and less deviation than Merlin.

Yeom Attack. To understand how the privacy noise influences Yeom attack success, we plot the loss distribution of member and non-member records for a private model trained with $\epsilon = 100$ in Figure 7a. The figure shows that the noise reduces the gap between the two distributions when compared to Figure 4a with no privacy. Hence differential privacy limits the success of Yeom by spreading out the loss values for both member and non-member distributions. This has the counter-productive impact of reducing the number of non-member records with zero loss from 959.2 ± 23.5 (in non-private case) to 98.0 ± 16.0. This reduces the minimum achievable false positive rate to 1%, and hence allows the attacker to set α thresholds smaller than 10% against private models which wasn't possible in the non-private case. However, the PPV is still less than 60% for these thresholds.

Figure 7b shows the attack performance at different thresholds. Due to the reduced gap between the member and non-member loss distributions, the PPV is close to 60% across all loss thresholds even if the maximum membership advantage is considerable (close to 20% for $\epsilon = 100$). Thus even with minimal privacy noise, the privacy leakage risk to membership inference attacks is significantly mitigated. For $\epsilon = 1$, the minimum false positive rate goes to 0.01%, allowing Yeom to achieve high PPV but with high deviation. The average PPV is close to 50%. We observe similar trends for other data sets and hence defer these results to Appendix C.

Merlin and Morgan Attacks. Figure 7c shows the distribution of Merlin ratio for member and non-member records on a private model trained with $\epsilon = 100$. When compared to the corresponding distribution for a non-private model (see Figure 5a), the gap between the distributions is greatly reduced. This restricts the privacy

[5] Due to the high cost of Shokri, we do not include it for this experiment. Although our experiments on Purchase-100X shows that Shokri does not pose significant privacy threat even for $\epsilon = 100$, where it achieves only 60% PPV.



(a) Loss distribution. **(b)** Yeom performance. **(c)** Merlin ratio distribution. **(d)** Merlin performance.

Fig. 7. Analysis of Yeom and Merlin on private model trained with $\epsilon = 100$ at $\gamma = 1$ (Purchase-100X).

leakage across all thresholds, as shown in Figure 7d. Though the maximum PPV can still be high enough to pose an exposure risk at higher privacy loss budgets. We observe similar trends for Merlin on the other data sets (see Appendix C). Unlike for non-private models, Morgan does not achieve close to 100% PPV as the members and non-members are not easily distinguishable due to the added privacy noise (see Figure 6b), but it does better than Merlin. Regardless, models trained with high privacy loss budgets can still be vulnerable to Merlin and Morgan even if Yeom does not succeed. This shows the importance of choosing appropriate privacy loss budgets for differential privacy mechanisms.

7.6 Imbalanced Scenarios

As discussed in Section 4, the membership advantage metric does not consider the prior distribution probability and hence does not capture the true privacy risk for imbalanced prior settings. In this section, we provide empirical evidence that the PPV metric captures privacy leakage more naturally in imbalanced prior settings, and hence is a more reliable privacy metric.

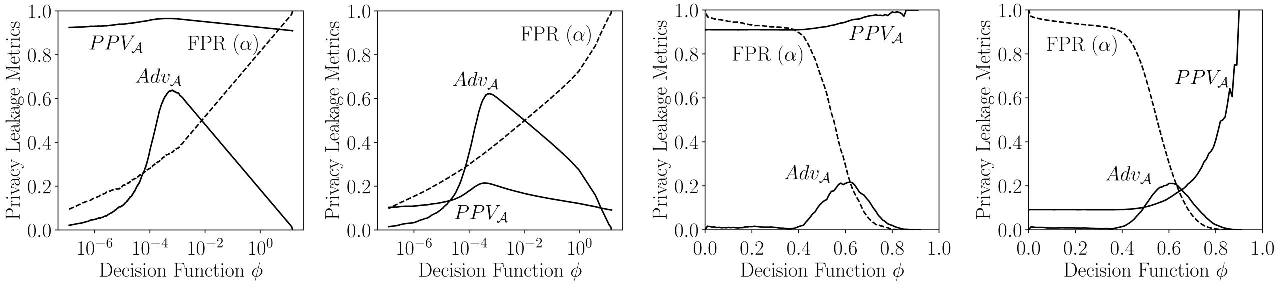
In imbalanced prior settings, the candidate pool from which the attacker samples records for inference testing has γ times more non-member records than members. In other words, a randomly selected candidate is γ times more likely to be a non-member than a member record. We keep the training set size fixed to 10,000 records as in our previous experiments, so need a test set size that is γ times the training set size. For each data set, we set γ as high as possible given the available data. As mentioned in Section 6, we constructed expanded versions of the Purchase-100 and RCV1 data sets to enable these experiments. Both the Purchase-100X and RCV1X data sets have more than 200,000 records, and hence are large enough to allow setting $\gamma = 10$. We did not have source data to expand Texas-100, so are left with a data set with only 67,000 records and hence only have results for $\gamma = 2$. The threshold selection procedure (Procedure 5.1) uses holdout training and test sets that are disjoint from the target training and test sets mentioned above, so the data set needs at least $(\gamma + 1) \times 20,000$ records to run the experiments.

	γ	Yeom	Merlin	Morgan
	0.1	96.5 ± 0.1	99.3 ± 0.7	100.0 ± 0.0
	0.5	84.5 ± 0.1	97.2 ± 2.8	100.0 ± 0.0
Purchase-100X	1.0	73.0 ± 0.2	93.4 ± 6.3	98.0 ± 4.0
	2.0	57.6 ± 0.3	84.0 ± 5.6	99.1 ± 1.7
	10.0	21.2 ± 0.1	69.7 ± 13.8	97.5 ± 5.0
	0.1	97.0 ± 0.1	99.2 ± 0.7	100.0 ± 0.0
	0.5	86.4 ± 1.1	95.0 ± 3.6	98.4 ± 0.5
Texas-100	1.0	76.1 ± 1.6	92.0 ± 4.5	95.7 ± 4.6
	2.0	62.4 ± 0.4	87.7 ± 11.1	97.4 ± 2.7
	10.0	-	-	-
	0.1	93.3 ± 0.5	99.8 ± 0.3	100.0 ± 0.0
	0.5	72.5 ± 0.9	94.3 ± 6.5	99.5 ± 1.0
RCV1X	1.0	57.9 ± 1.0	98.8 ± 2.4	100.0 ± 0.0
	2.0	40.5 ± 1.5	98.8 ± 2.4	98.8 ± 2.4
	10.0	12.2 ± 0.3	74.3 ± 15.9	93.0 ± 9.8
	0.1	96.0 ± 0.2	97.9 ± 1.9	100.0 ± 0.0
	0.5	84.6 ± 0.5	86.4 ± 1.7	100.0 ± 0.0
CIFAR-100	1.0	72.7 ± 0.8	75.0 ± 2.6	100.0 ± 0.0
	2.0	56.7 ± 0.6	74.0 ± 8.1	100.0 ± 0.0
	10.0	-	-	-

Table 6. Effect of varying γ on maximum PPV achieved by attacks against non-private models. All values are in percentage.

Table 6 shows the effect of varying γ on the maximum PPV of inference attacks against non-private models trained on different data sets. We can see a clear drop in PPV values across all data sets with increasing γ values for Yeom and Merlin.[6] Although, Merlin consistently outperforms Yeom across all settings. At $\gamma = 0.1$, the

[6] Due to the high cost of Shokri, we do not include it for this experiment.



(a) Yeom performance at $\gamma = 0.1$. **(b)** Yeom performance at $\gamma = 10$. **(c)** Merlin performance at $\gamma = 0.1$. **(d)** Merlin performance at $\gamma = 10$.

Fig. 8. Attack performance on Purchase-100X for imbalanced prior setting. $PPV_{\mathcal{A}}$ varies with γ, while $Adv_{\mathcal{A}}$ remains almost same.

base rate for PPV is 90%. While Yeom achieves around 96% PPV, Merlin achieves close to 100% PPV across all data sets. For $\gamma = 2$, the maximum PPV of Yeom is close to 60%, whereas Merlin still achieves high enough PPV to pose some privacy threat. Both the Yeom and Merlin are less successful as the γ value increases to 10. However, Morgan consistently achieves close to 100% PPV across all settings, thereby showing the vulnerability of non-private models even in the skewed prior settings. This is graphically shown in Figure 6c where Morgan is able to identify the most vulnerable members on Purchase-100X even at $\gamma = 10$. The advantage values remain more or less the same across different γ values for both Yeom and Merlin on Purchase-100X, as shown in Figure 8. These results support our claim that PPV is a more reliable metric in skewed prior scenarios. We observe the same trend for the other data sets, and hence do not include their plots due to space limit.

While Yeom and Merlin do not pose an exposure threat in the imbalanced prior settings where γ values are higher than 10, Morgan still exposes some vulnerable members with close to 100% PPV. Thus, our proposed attacks pose significant threat even in more realistic settings of skewed priors, where the existing attacks fail. We observe that the private models are not vulnerable to any of our inference attacks in the imbalanced prior setting where $\gamma > 1$. At $\gamma = 2$, the best attack achieves maximum PPV close to 48% across all data sets, whereas at $\gamma = 10$, this further drops to around 17%. Hence we do not show the membership inference attack results against private models for these settings.

8 Conclusion

Understanding the privacy risks posed by machine learning involves considerable challenges, and there remains a large gap between achievable privacy guarantees, and what can be inferred using known attacks in practice. While membership inference has previously been evaluated in balanced prior settings, we consider scenarios with imbalanced priors and show that there are attacks which pose serious privacy threats even in such settings where previous attacks fail.

We introduce a novel threshold selection procedure that allows adversaries to choose inference thresholds specific to their attack goals, and propose two new membership inference attacks, Merlin and Morgan, that outperform previous attacks in the settings that concern us most: being able to identify members, with very high confidence, even from candidate pools where most candidates are not members. From experiments on four data sets under different prior distribution settings, we find that the non-private models are highly vulnerable to such attacks, and the models trained with high privacy loss budgets can still be vulnerable.

Availability

All of our code and data for our experiments is available at https://github.com/bargavj/EvaluatingDPML.

Acknowledgments

We would like to thank Esfandiar Mohammadi for shepherding our paper and for providing valuable feedback to improve the paper. This work was partially supported by grants from the National Science Foundation (#1717950 and #1915813).

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, 2019.

[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] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. *arXiv:1905.09982*, 2019.

[5] 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.

[6] 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.

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

[8] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 2011.

[9] Christopher A Choquette Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. *arXiv:2007.14321*, 2020.

[10] Kaggle Competition. Acquire valued shoppers challenge, 2014.

[11] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv:1905.02383*, 2019.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*, 2006.

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

[14] Farhad Farokhi and Mohamed Ali Kaafar. Modelling and quantifying membership information leakage in machine learning. *arXiv:2001.10648*, 2020.

[15] 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.

[16] 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*, 2014.

[17] 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.

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

[19] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 2008.

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

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

[22] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.

[23] 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.

[24] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *CCS*, 2019.

[25] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 2017.

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[27] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 2004.

[28] Changchang Liu, Xi He, Thee Chanyaswad, Shiqiang Wang, and Prateek Mittal. Investigating statistical privacy frameworks from the perspective of hypothesis testing. *Proceedings on Privacy Enhancing Technologies*, 2019.

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

[30] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv:1802.04889*, 2018.

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

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

[33] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning. In *IEEE Symposium on Security and Privacy*, 2019.

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

[35] 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*, 2019.

[36] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 2009.

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

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

[39] Congzheng Song. Code for membership inference attack against machine learning models. https://github.com/csong27/membership-inference, 2017.

[40] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. *arXiv:2003.10595*, 2020.

[41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

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

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

[44] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 2010.

[45] Mengjia Yan, Christopher Fletcher, and Josep Torrellas. Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. In *USENIX Security Symposium*, 2020.

[46] 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.

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

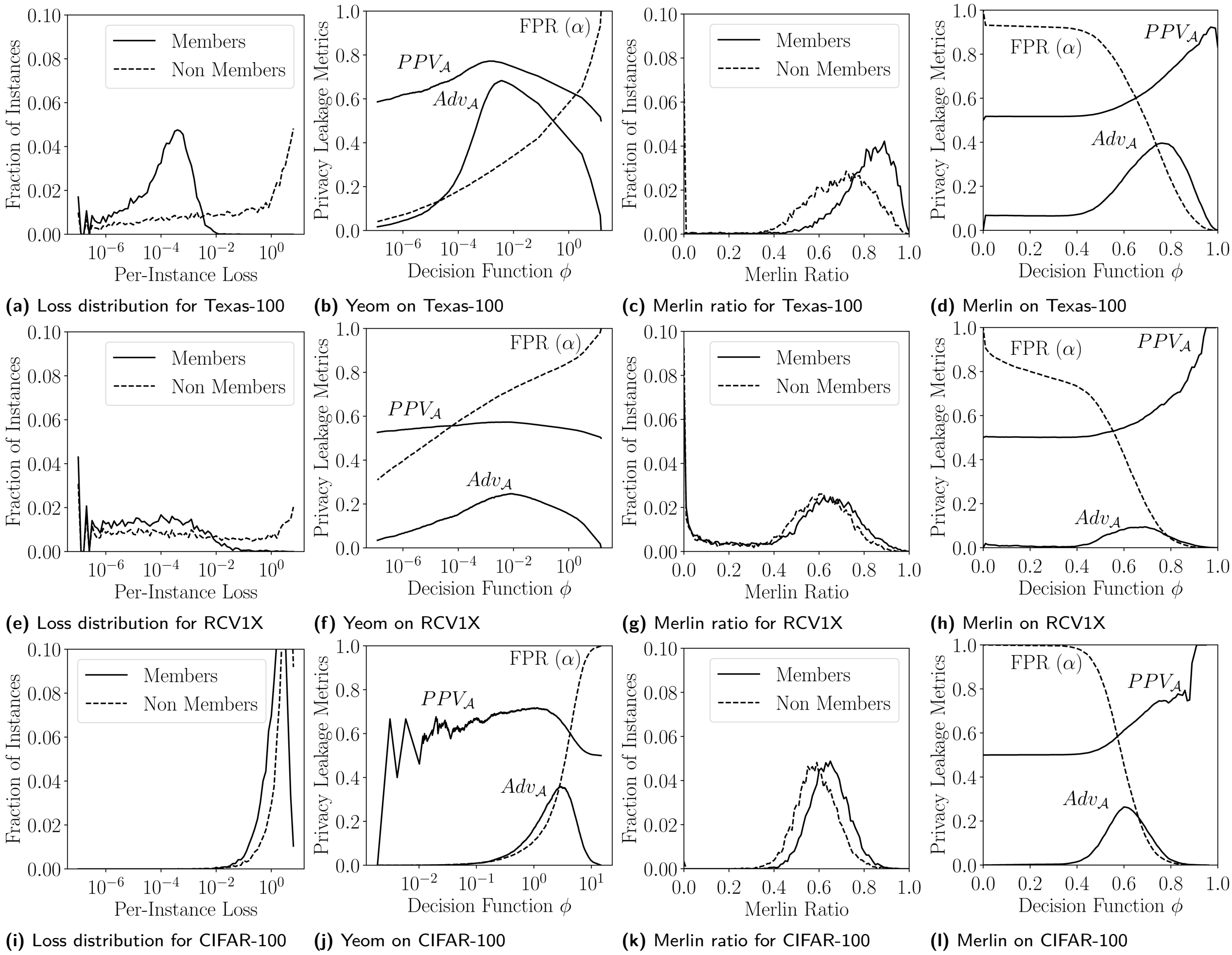
A Hyperparameters

For each data set, the training set is fixed to 10,000 randomly sampled records and the test set size is varied to reflect different prior probability distributions. We sample γ times the number of training set records to create the test set. For both Purchase-100X and RCV1X, we use $\gamma = \{0.1, 0.5, 1, 2, 10\}$; for Texas-100 and CIFAR-100, we use $\gamma = \{0.1, 0.5, 1, 2\}$ since they are too small for experiments with larger γ. For training the models, we use the Adam optimizer and perform grid search to find the best values for hyperparameters such as batch size, learning rate, ℓ_2 penalty, clipping threshold and number of iterations. We find a batch size of 200, clipping threshold of 4, and ℓ_2 penalty of 10^{-8} work best across all the data sets, except for CIFAR-100 where we use ℓ_2 penalty of 10^{-4}, and RCV1X where we use clipping threshold of 1. We use a learning rate of 0.005 for Purchase-100X and Texas-100, 0.003 for RCV1X, and 0.001 for CIFAR-100. We set the training epochs to 100 for Purchase-100X and CIFAR-100, 30 for Texas-100, and 80 for RCV1X. We fix the differential privacy failure parameter δ as 10^{-5} to keep it smaller than the inverse of the training set size, generally considered the maximum acceptable δ value.

B Additional Results for Non-Private Models

Results on Texas-100. We plot the distribution of per-instance loss for a non-private model trained on Texas-100 in Figure 9a. A notable difference is that the number of non-members having zero loss is lower than that of Purchase-100X. As a result, the false positive rate can be as low as 3% for this data set. This is depicted in Figure 9b which shows the performance of Yeom against a non-private model at different thresholds. The trend is similar to what we observe for Purchase-100X.

Figure 9c shows the distribution of Merlin ratio against a non-private model trained on Texas-100. The gap between the member and non-member distributions is greater than that of Purchase-100X and hence this attack is more effective on this data set. An important indicator is that all members have non-zero Merlin ratio. The average Merlin ratio is 0.81 ± 0.12 for members whereas it is 0.65 ± 0.22 for non-members. Figure 9d shows the performance of Merlin on non-private model at different count thresholds. These results further validate the effectiveness of selecting a good threshold based on our proposed procedure. Figure 10a shows the scatter plot of per-instance loss and Merlin ratio for all records. Similar to the case of Purchase-100X, more fraction of members are concentrated between 1.2×10^{-4} and 5.1×10^{-3} loss and have Merlin ratio greater than 0.90. Table 7 compares the membership inference attacks across different attack settings on Texas-100. As shown, Merlin achieves much higher PPV values than Yeom. Using class based thresholds drastically improves PPV for Yeom such that Yeom CBT achieves maximum PPV comparable to Merlin. As with Purchase-100X, we observe no benefit of using CBT for Merlin. While Shokri



(a) Loss distribution for Texas-100 **(b)** Yeom on Texas-100 **(c)** Merlin ratio for Texas-100 **(d)** Merlin on Texas-100

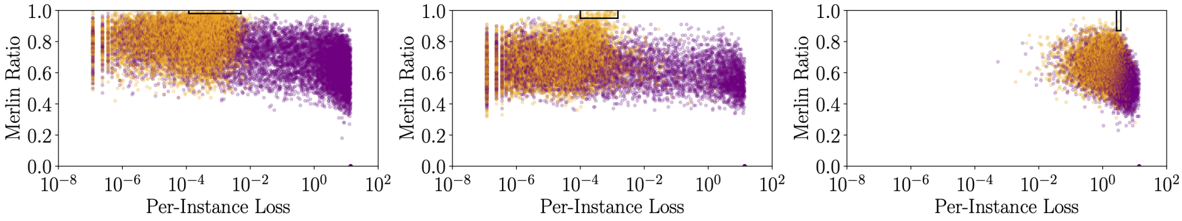
(e) Loss distribution for RCV1X **(f)** Yeom on RCV1X **(g)** Merlin ratio for RCV1X **(h)** Merlin on RCV1X

(i) Loss distribution for CIFAR-100 **(j)** Yeom on CIFAR-100 **(k)** Merlin ratio for CIFAR-100 **(l)** Merlin on CIFAR-100

Fig. 9. Analysis of Yeom and Merlin against non-private models trained on different data sets in the balanced prior setting.

		α	ϕ	Actual FPR	Actual TPR	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$
Yeom	**Fixed FPR**	1.00	-	-	-	-	-
	Min FPR	3.00	0	0.8 ± 1.7	1.3 ± 2.6	0.4 ± 0.9	12.0 ± 24.0
	Fixed ϕ	-	$(1.1 \pm 2.0) \times 10^{-2}$	26.1 ± 5.0	77.5 ± 7.4	51.3 ± 2.6	75.0 ± 1.6
	Max $PPV_{\mathcal{A}}$	26.00	$(1.8 \pm 0.4) \times 10^{-3}$	26.6 ± 0.3	85.8 ± 14.5	59.2 ± 11.7	76.1 ± 1.6
	Max $Adv_{\mathcal{A}}$	31.00	$(6.6 \pm 1.3) \times 10^{-3}$	31.4 ± 2.8	94.3 ± 10.3	62.9 ± 7.7	75.0 ± 0.6
Yeom CBT	**Min FPR**	0.01	$0,\ 3.4 \times 10^{-6},\ 9.2 \times 10^{-2}$	1.0 ± 0.5	11.2 ± 3.1	10.2 ± 2.6	92.0 ± 2.3
	Max $PPV_{\mathcal{A}}$	0.01	$0,\ 3.4 \times 10^{-6},\ 9.2 \times 10^{-2}$	1.0 ± 0.5	11.2 ± 3.1	10.2 ± 2.6	92.0 ± 2.3
	Fixed FPR	1.00	$0,\ 4.8 \times 10^{-6},\ 9.2 \times 10^{-2}$	1.3 ± 0.6	12.5 ± 3.4	11.2 ± 2.9	91.2 ± 2.1
	Fixed ϕ	-	$(0.1,\ 8.3,\ 554.8) \times 10^{-4}$	21.7 ± 3.6	70.9 ± 15.7	49.2 ± 12.1	76.3 ± 1.4
	Max $Adv_{\mathcal{A}}$	52.00	$1.2 \times 10^{-7},\ 7.3 \times 10^{-3},\ 4.3$	28.9 ± 4.2	90.4 ± 11.5	61.5 ± 8.0	75.8 ± 1.2
Merlin	**Min FPR**	0.01	1.00 ± 0.01	0.0 ± 0.0	0.1 ± 0.1	0.1 ± 0.1	51.9 ± 42.4
	Max $PPV_{\mathcal{A}}$	0.06	0.99 ± 0.00	0.0 ± 0.0	0.3 ± 0.2	0.3 ± 0.2	92.0 ± 4.5
	Fixed FPR	1.00	0.95 ± 0.00	0.8 ± 0.2	5.7 ± 1.4	4.9 ± 1.3	87.8 ± 2.7
	Max $Adv_{\mathcal{A}}$	36.00	0.76 ± 0.00	33.6 ± 1.1	71.4 ± 1.2	37.8 ± 1.5	68.0 ± 0.8
Morgan	**Max $PPV_{\mathcal{A}}$**	-	$1.2 \times 10^{-4},\ 5.1 \times 10^{-3},\ 0.98$	0.0 ± 0.0	0.6 ± 0.2	0.5 ± 0.2	95.7 ± 4.6

Table 7. Thresholds selected against non-private models trained on Texas-100 with balanced prior. The results are averaged over five runs such that the target model is trained from the scratch for each run. Yeom CBT uses class-based thresholds, where ϕ shows the triplet of minimum, median and maximum thresholds across all classes. All values, except ϕ, are reported in percentage.



(a) Texas-100 **(b)** RCV1X **(c)** CIFAR-100

Fig. 10. Comparing loss and Merlin ratio for non-private models trained on different data sets at $\gamma = 1$. Members and non-members are denoted by orange and purple points respectively. Highlighted boxes denote the members identified by Morgan at max PPV.

		α	ϕ	Actual FPR	Actual TPR	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$
Yeom	**Fixed FPR**	1.00	-	-	-	-	-
	Min FPR	33.00	0	7.4 ± 14.8	7.8 ± 15.6	0.4 ± 0.8	10.3 ± 20.5
	Max $PPV_{\mathcal{A}}$	67.00	$(0.5 \pm 0.2) \times 10^{-3}$	65.7 ± 3.9	90.6 ± 8.8	24.8 ± 5.0	57.9 ± 1.0
	Max $Adv_{\mathcal{A}}$	70.00	$(1.5 \pm 0.6) \times 10^{-3}$	68.9 ± 3.2	94.1 ± 6.2	25.1 ± 3.2	57.7 ± 0.6
	Fixed ϕ	-	$(3.2 \pm 3.2) \times 10^{-3}$	70.2 ± 1.0	97.1 ± 2.2	26.9 ± 2.7	58.0 ± 0.8
Yeom CBT	**Min FPR**	0.01	$0, 0, 3.8 \times 10^{-3}$	0.1 ± 0.1	1.3 ± 0.3	1.2 ± 0.3	93.1 ± 3.2
	Max $PPV_{\mathcal{A}}$	0.01	$0, 0, 3.8 \times 10^{-3}$	0.1 ± 0.1	1.3 ± 0.3	1.2 ± 0.3	93.1 ± 3.2
	Fixed FPR	1.00	$0, 2.4 \times 10^{-8}, 3.8 \times 10^{-3}$	0.1 ± 0.1	1.4 ± 0.3	1.3 ± 0.3	92.7 ± 3.5
	Max $Adv_{\mathcal{A}}$	70.00	$0, 1.4 \times 10^{-3}, 9.0$	50.9 ± 7.6	73.3 ± 10.5	22.4 ± 3.2	59.0 ± 0.5
	Fixed ϕ	-	$(0.1, 2.8, 91.1) \times 10^{-4}$	62.4 ± 4.8	84.0 ± 10.2	21.6 ± 5.6	57.3 ± 1.2
Merlin	**Min FPR**	0.01	0.97 ± 0.01	0.0 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	98.8 ± 2.4
	Max $PPV_{\mathcal{A}}$	0.01	0.97 ± 0.01	0.0 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	98.8 ± 2.4
	Fixed FPR	1.00	0.88 ± 0.00	0.7 ± 0.2	3.3 ± 0.7	2.6 ± 0.7	81.7 ± 4.3
	Max $Adv_{\mathcal{A}}$	26.00	0.66 ± 0.00	24.9 ± 2.0	36.5 ± 1.6	11.6 ± 2.3	59.5 ± 2.0
Morgan	**Max $PPV_{\mathcal{A}}$**	-	$1.0 \times 10^{-4}, 1.5 \times 10^{-3}, 0.95$	0.0 ± 0.0	0.4 ± 0.3	0.4 ± 0.3	100.0 ± 0.0

Table 8. Thresholds selected against non-private models trained on RCV1X with balanced prior. The results are averaged over five runs such that the target model is trained from the scratch for each run. Yeom CBT uses class-based thresholds, where ϕ shows the triplet of minimum, median and maximum thresholds across all classes. All values, except ϕ, are reported in percentage.

achieves 89% PPV, slightly less than Merlin, on this data set (see Table 4), using CBT decreases the PPV to 85%. Morgan achieves highest PPV among all attacks.

Results on RCV1X. We plot the per-instance loss distribution for a non-private model trained on RCV1X in Figure 9e. While more members are concentrated closer to zero loss than the non-members, we observe that the gap between the two distributions is not as large as with the other data sets. Moreover, 3504 ± 444 non-members have zero loss, and hence the minimum false positive rate for Yeom is around 33%. Figure 9f shows the performance of Yeom for different loss thresholds. The maximum PPV that can be achieved is only around 58%, at which point the advantage is close to 27%. Thus while the advantage metric would suggest that there is privacy risk, Yeom does not pose significant risk. Shokri, on the other hand, achieves a PPV of 91% (see Table 4) and poses a significant privacy risk.

Figure 9g shows the distribution of Merlin ratio for a non-private model trained on RCV1X. While the gap between distributions is small, the PPV can still be high as depicted in Figure 9h. Merlin achieves a maximum PPV of around 99% on an average for threshold values close to 0.97, and hence poses privacy threat even when Yeom fails. Table 8 compares the attacks on RCV1X for different attack goals. Yeom is benefited from using class based thresholds, as the maximum PPV jumps from 58% to 93%. However, Merlin still outperforms Yeom CBT at maximum PPV setting. As with other data sets, Shokri does not benefit from CBT technique. Figure 10b shows the loss and Merlin ratio scatter plot on RCV1X. Though the members and non-members are less differentiated, Morgan is still able to identify the most vulnerable members with 100% confidence (see Table 8).

Results on CIFAR-100. Figure 9i shows the distribution of per-instance loss for a non-private model trained on CIFAR-100. The loss of both members and non-members is high, since the model does not completely overfit on this data set. Figure 9j shows the performance of Yeom for different loss thresholds. Figures 9k and 9l show the distribution of Merlin ratio and leakage metrics for different thresholds. Using CBT on Yeom increases

		α	ϕ	Actual FPR	Actual TPR	$Adv_{\mathcal{A}}$	$PPV_{\mathcal{A}}$
Yeom	Min FPR	0.01	$(4.3 \pm 2.1) \times 10^{-3}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	33.3 ± 27.9
	Fixed FPR	1.00	$(7.2 \pm 0.8) \times 10^{-2}$	0.8 ± 0.1	1.4 ± 0.4	0.7 ± 0.3	65.1 ± 2.6
	Max $PPV_{\mathcal{A}}$	12.00	1.0 ± 0.0	11.4 ± 0.4	30.4 ± 1.9	19.0 ± 1.6	72.7 ± 0.8
	Fixed ϕ	-	2.0 ± 0.1	24.0 ± 1.0	57.0 ± 0.5	33.0 ± 1.6	70.3 ± 1.1
	Max $Adv_{\mathcal{A}}$	39.00	2.9 ± 0.0	37.9 ± 0.5	75.1 ± 1.5	37.2 ± 1.8	66.5 ± 0.6
Yeom CBT	Min FPR	0.01	0, 0.1, 1.8	1.0 ± 0.2	4.4 ± 1.1	3.4 ± 0.9	81.2 ± 2.3
	Max $PPV_{\mathcal{A}}$	0.01	0, 0.1, 1.8	1.0 ± 0.2	4.4 ± 1.1	3.4 ± 0.9	81.2 ± 2.3
	Fixed FPR	1.00	0, 0.2, 2.0	1.6 ± 0.2	6.6 ± 0.9	5.1 ± 0.8	81.0 ± 1.4
	Fixed ϕ	-	0.7, 2.0, 3.2	22.5 ± 1.1	56.6 ± 3.5	34.0 ± 2.7	71.5 ± 0.7
	Max $Adv_{\mathcal{A}}$	40.00	0.5, 3.0, 4.6	37.9 ± 0.4	75.4 ± 1.1	37.5 ± 1.5	66.6 ± 0.6
Merlin	Min FPR	0.01	0.92 ± 0.02	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	51.4 ± 32.0
	Max $PPV_{\mathcal{A}}$	0.90	0.82 ± 0.00	0.8 ± 0.2	2.3 ± 0.6	1.6 ± 0.5	75.0 ± 2.6
	Fixed FPR	1.00	0.82 ± 0.00	0.9 ± 0.2	2.8 ± 0.7	1.8 ± 0.5	74.7 ± 1.7
	Max $Adv_{\mathcal{A}}$	39.00	0.62 ± 0.00	38.1 ± 1.1	65.8 ± 2.3	27.7 ± 1.3	63.3 ± 0.3
Morgan	Max $PPV_{\mathcal{A}}$	-	2.7, 3.7, 0.87	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	100.0 ± 0.0

Table 9. Thresholds selected against non-private models trained on CIFAR-100 with balanced prior. The results are averaged over five runs such that the target model is trained from the scratch for each run. Yeom CBT uses class-based thresholds, where ϕ shows the triplet of minimum, median and maximum thresholds across all classes. All values, except ϕ, are percentages.

		Yeom			Merlin			Morgan	
	ϵ	α	ϕ	Max $PPV_{\mathcal{A}}$	α	ϕ	Max $PPV_{\mathcal{A}}$	ϕ	Max $PPV_{\mathcal{A}}$
Texas-100	1	0.05	$(1.0 \pm 0.5) \times 10^{-2}$	58.5 ± 4.6	0.13	0.92	61.0 ± 5.4	0.3, 1.4, 0.93	85.6 ± 19.8
	10	0.06	$(1.0 \pm 0.4) \times 10^{-5}$	65.5 ± 19.8	0.05	0.94	67.2 ± 19.3	5.4×10^{-2}, 0.2, 0.93	76.7 ± 26.1
	100	0.02	$(0.2 \pm 0.2) \times 10^{-5}$	58.8 ± 24.0	0.45	0.92	59.5 ± 3.6	0, 8.4×10^{-5}, 0.92	68.2 ± 17.9
RCV1X	1	13.00	$(6.0 \pm 1.1) \times 10^{-4}$	51.7 ± 0.5	3.00	0.80	52.6 ± 2.0	0, 5.0×10^{-6}, 0.80	75.0 ± 21.1
	10	60.00	$(2.4 \pm 0.3) \times 10^{-2}$	51.8 ± 0.2	0.10	0.89	70.9 ± 12.9	4.7×10^{-5}, 3.6, 0.89	77.4 ± 11.0
	100	70.00	$(3.7 \pm 0.3) \times 10^{-2}$	53.0 ± 0.1	0.04	0.92	86.9 ± 11.6	2.7×10^{-5}, 12, 0.92	89.1 ± 11.3
CIFAR-100	1	0.11	4.3 ± 0.0	68.4 ± 27.1	0.11	0.85	57.3 ± 10.1	4.5, 4.8, 0.85	62.7 ± 7.7
	10	0.11	1.2 ± 0.1	64.2 ± 29.4	0.07	0.79	66.6 ± 17.6	0.7, 3.1, 0.79	77.3 ± 12.6
	100	0.80	1.3 ± 0.0	56.3 ± 3.2	0.12	0.77	62.0 ± 11.5	1.4, 2.2, 0.77	89.7 ± 13.5

Table 10. MI attacks against private models trained on different data sets in the balanced prior setting. α and PPV values are in percentage. Merlin's ϕ has 0 standard deviation for Texas-100, and ±0.01 standard deviation for RCV1X and CIFAR-100.

the maximum PPV from 73% to 81% (Table 9), exceeding that of Merlin. Shokri is less successful on this data set, achieving only 65% PPV (Table 4), and does not benefit from the CBT technique. Figure 10c shows the loss and Merlin ratio of all records on CIFAR-100. As shown, members with high Merlin ratio are distinguishable from non-members. Morgan is able to identify certain members with 100% PPV (see Table 9).

C Additional Privacy Results

The plots for private models on all three data sets are similar to that of Purchase-100X, hence we do not include them here. Instead, we directly compare the maximum PPV of the attacks against private models trained with varying privacy loss budgets across all three data sets in Table 10. As with Purchase-100X, adding noise allows Yeom to set much smaller thresholds on Texas-100. For higher ϵ values, Yeom poses some privacy threat but the PPV deviation is high. On RCV1X, the α values are still high and hence Yeom is not successful even for $\epsilon = 100$. On CIFAR-100, Yeom achieves considerably higher PPV values for $\epsilon = 1$ and $\epsilon = 10$, but the deviation is very high, indicating that the attack is only successful for some runs. At $\epsilon = 100$, Yeom fails to pose any threat. Merlin achieves higher PPV than Yeom on average across all data sets. Similar to Purchase-100X, Merlin achieves high PPV values for $\epsilon = 10$ and $\epsilon = 100$ on RCV1X. However, it does not achieve high enough PPV on Texas-100 and CIFAR-100 to pose a serious privacy threat, even for $\epsilon = 100$. On the other hand, Morgan poses serious privacy threat against models trained with high privacy loss budgets across all tested data sets.