Antipodes of Label Differential Privacy: PATE and ALIBI

Mani Malek* **Ilya Mironov*** **Karthik Prasad*** **Igor Shilov*** **Florian Tramèr†**

Abstract

We consider the privacy-preserving machine learning (ML) setting where the trained model must satisfy differential privacy (DP) with respect to the *labels* of the training examples. We propose two novel approaches based on, respectively, the Laplace mechanism and the PATE framework, and demonstrate their effectiveness on standard benchmarks.

While recent work by Ghazi et al. proposed Label DP schemes based on a randomized response mechanism, we argue that additive Laplace noise coupled with Bayesian inference (ALIBI) is a better fit for typical ML tasks. Moreover, we show how to achieve very strong privacy levels in some regimes, with our adaptation of the PATE framework that builds on recent advances in semi-supervised learning.

We complement theoretical analysis of our algorithms' privacy guarantees with empirical evaluation of their memorization properties. Our evaluation suggests that comparing different algorithms according to their *provable* DP guarantees can be misleading and favor a less private algorithm with a tighter analysis.

Code for implementation of algorithms and memorization attacks is available from `https://github.com/facebookresearch/label_dp_antipodes`.

1 Introduction

Sophisticated machine learning models perform well on their target tasks despite of—or thanks to—their predilection for data memorization [39, 2, 17]. When such models are trained on non-public inputs, privacy concerns become paramount [31, 6, 7], which motivates the actively developing area of privacy-preserving machine learning.

With some notable exceptions, which we discuss in the related works section, existing research has focused on an "all or nothing" privacy definition, where all of the training data, i.e., both *features* and *labels*, is considered private information. While this goal is appropriate for many applications, there are several important scenarios where *label-only* privacy is the right solution concept.

A prominent example of label-only privacy is in online advertising, where the goal is to predict conversion of an ad impression (the label) given a user's profile and the spot's context (the features). The features are available to the advertising network, which trains the model, while the labels (data on past conversion events) are visible only to the advertiser. More generally, any two-party setting where data is vertically split between public inputs and (sensitive) outcomes is a candidate for training with label-only privacy.

In this work we adopt the definition of differential privacy as our primary notion of privacy. The definition, introduced in the seminal work by Dwork et al. [13], satisfies several important properties that make it well-suited for applications: composability, robustness to auxiliary information, preservation under post-processing, and graceful degradation in the presence of correlated inputs (group privacy). Following Ghazi et al. [18], we refer to the setting of label-only differential privacy as Label DP.

*{manimalek,imironov,krp,shilov}@fb.com, Facebook AI.

†tramer@cs.stanford.edu, Stanford University.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Our contributions. We explore two approaches—with very different characteristics—toward Label DP. These approaches and our methodology for estimating empirical privacy loss via label memorization are briefly reviewed below:

- *PATE*: We adapt the PATE framework (Private Aggregation of Teacher Ensembles) of Papernot et al. [28] to the Label DP setting by observing that PATE's central assumption—availability of a public unlabeled dataset—can be satisfied for free. Indeed, a public unlabeled dataset is simply the private dataset with the labels *removed*! By instantiating PATE with a state-of-the-art technique for semi-supervised learning, we demonstrate an excellent tradeoff between empirical privacy loss and accuracy.
- *ALIBI (Additive Laplace with Iterative Bayesian Inference)*: We propose applying additive Laplace noise to a one-hot encoding of the label. Since differential privacy is preserved under post-processing, we can de-noise the mechanism's output by performing Bayesian inference using the prior provided by the model itself, doing so continuously during the training process. ALIBI improves, particularly in a high-privacy regime, Ghazi et al.'s approach based on randomized response [18].
- *Empirical privacy loss.* We describe a memorization attack targeted at the Label DP scenario that efficiently extracts lower bounds for the privacy parameter that, in some cases, come within a factor of 2.5 from the theoretical, worst-case upper bounds against practically relevant, large-scale models. For comparison, recent, most advanced black-box attacks on DP-SGD have approximately a $10\times$ gap between the upper and lower bounds [20, 26].

Both algorithms, PATE and ALIBI, come with rigorous privacy analyses that, for any fixed setting of parameters, result in an (ε, δ)-DP bound. We emphasize that these bounds are just that—they are upper limits on an attacker's ability to breach privacy (in a precise sense) on worst-case inputs. Even though these bounds can be pessimistic, intuitively, a smaller ε for the same δ corresponds to a more private instantiation of a mechanism, and indeed this is likely the case within an algorithmic family.

Can the privacy upper bounds be used to compare diverse mechanisms with widely dissimilar analyses, such as ours? We argue that focusing on the upper bounds alone may be misleading to the point of favoring a less private algorithm with tighter analysis over a more private one whose privacy analysis is less developed. For a uniform perspective, we estimate the *empirical* privacy loss of all trained models by evaluating the performance of a black-box membership inference attack [31, 20, 26].

Our attack directly measures memorization of deliberately mislabeled examples, or *canaries* [6]. Following prior work [26, 20], we use the success rate of our attack to compute a lower bound on the level ε of label-DP achieved by the training algorithm. To avoid the prohibitive cost of retraining a model for each canary, we propose and analyze a heuristic that consists in inserting multiple canaries into one model, and measuring an attacker's success rate in guessing each canary's label. In some settings, we find that the empirical privacy of PATE (as measured by our attack) is significantly stronger than that of ALIBI, even though the provable DP bound for PATE is much weaker.

Finally, to characterize setups where label-only privacy may be applicable, it is helpful to distinguish between *inference* tasks, where the label is fully determined by the features, and *prediction*, where there is some uncertainty as to the eventual outcome, despite (hopefully) some predictive value of the feature set. We are mostly concerned with the latter scenario, where the outcome is not evident from the features, or protecting choices made by individuals may be mandated or desirable (for connection between privacy and self-determination see Schwartz [30]). While we do use public data sets (CIFAR-10 and CIFAR-100) in which it is possible to "infer" the label, we do so only to evaluate our approaches on standard benchmarks. We completely remove the triviality of *inference* by adding mislabeled canaries in our attack, thereby introducing a measure of non-determinism.

2 Notation and Definitions

Throughout the paper we consider standard supervised learning problems. The inputs are (x, y) pairs where $x \in X$ are the *features* and $y \in Y$ are the *labels*. (If some labels are absent, the problem becomes semi-supervised learning.) The cardinality of the set Y is the number of *classes* C. The task is to learn a model M parameterized with weights W that predicts y given x. To this end the training algorithm has access to a training set of n samples $\{(x_i, y_i)\}_n$.

Differential privacy due to Dwork et al. [13] is a rigorous, quantifiable notion of individual privacy for statistical algorithms. (See Dwork and Roth [14] and Vadhan [33] for book-length introductions.)

Definition 2.1 (Differential Privacy)**.** *We say that a randomized mechanism $\mathcal{M}\colon U \to R$ satisfies (ε, δ)-differential privacy (DP) if for all* adjacent *inputs $D, D' \in U$ that differ in contributions of a single sample, the following holds:*

$$\forall S \subset R \quad \Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in S] + \delta.$$

If $\delta = 0$, we refer to it as ε-DP.

The notion of adjacency between databases D and D' is domain- and application-dependent. To instantiate the Label DP setting, we say that D and D' are adjacent if they consist of the same number of examples $\{(x_i, y_i)\}_n$ and are identical except for a difference in a single label at index i^*: $(x_{i^*}, y_{i^*}) \in D$ and $(x_{i^*}, y'_{i^*}) \in D'$.

3 Background and related work

DP and machine learning. Differential privacy aligns extremely well with the goal of machine learning, which is to produce valid models or hypotheses on the basis of observed data. In fact, the relationship can be made precise. An important metric in ML is *generalization error*, which measures the model's performance on previously unseen samples. Differentially private mechanisms provably generalize [11]. For an introduction to some important early results on connections between differential privacy and learning theory see Dwork and Roth [14, Chapter 11].

Randomized response (RR). As a disclosure avoidance methodology, randomized response (RR) [36] precedes the advent of differential privacy. In RR's basic form, with some probability the sensitive input is replaced by a sample drawn uniformly from its entire domain. RR can be implemented at the source, making it an essential building block in achieving Local DP [16].

More generally, RR is an instance of the *input perturbation* technique, which leaves the ML training algorithm intact and achieves differential privacy via input randomization. Input perturbations methods are particularly friendly to implementers as they require minimal changes to existing algorithms, treating them as black boxes.

Additive noise mechanisms. Another important class of differentially private mechanisms are *additive noise* methods. Two canonical examples are additive Laplace [13] and additive Gaussian [12] that can be applied to vector-valued functions. Their noise distributions are calibrated to the function's *sensitivity*—the maximal distance between the function's outputs on adjacent inputs (D and D' in Definition 2.1) under the ℓ_1 or ℓ_2 metrics for the Laplace and Gaussian mechanisms respectively.

PATE. Private Aggregation of Teacher Ensembles (PATE) [28, 29] is a framework based on private knowledge aggregation of an ensemble model and knowledge transfer. PATE trains an ensemble of "teachers" on disjoint subsets of the private dataset. The ensemble's knowledge is then transferred to a "student" model via differentially private aggregation of the teachers' votes on samples from an unlabeled public dataset. Only the student model is released as the output of the training, as it accesses sensitive data via a privacy-preserving interface.

Since PATE consumes the privacy budget each time the student queries the teachers, the training algorithm must be optimized to minimize the number of queries. To this end, Papernot et al. explore the use of semi-supervised learning techniques, most notably GANs. Recent advances in semi-supervised learning have been shown to boost the performance of PATE on standard benchmarks [5].

FixMatch. FixMatch [32] is a state-of-the-art semi-supervised learning algorithm, which achieves high accuracy on benchmark image datasets with very few labeled examples. It is based on the concepts of pseudo-labeling [23] and consistency regularization [3]: FixMatch uses a partially trained model to generate labels on weakly-augmented unlabeled data, and then, if the prediction is confident enough, uses the output as a label for the strongly-augmented version of the same data point.

Label-only privacy. The setting of label-only DP was formally introduced by Chaudhuri and Hsu [8], who proved lower bounds on sample complexity for private PAC-learners. Beimel et al. [4] demonstrated that the sample complexity of Label DP matches that of non-private PAC learning (ignoring constants, dependencies on the privacy parameters, and computational efficiency). Wang and Xu [34] considered the linear regression problem in the local, label-only privacy setting.

Most recently, Ghazi et al. [18] considered label-only privacy for deep learning. They propose a Label Privacy Multi-Stage Training (LP-MST) algorithm, which we revisit in Section 5. The algorithm advances in stages (Ghazi et al. evaluate it up to four), training the model on disjoint partitions of the dataset. In each stage, the private labels are protected using a randomized response scheme.

The most innovative part of LP-MST happens at the point of transition between stages. Labels of the training examples of the next partition are perturbed using randomized response. The (soft) output of the previous stage's model is interpreted as a prior, which is used to choose parameters for the randomized response algorithm *separately for each example*. For instance, if the current model assigns high probabilities to labels 1–10 and low probabilities to labels 11–100, the randomized response will be constrained to outputting a label within the set 1–10.

We build on the insights of Ghazi et al. by making two observations. First, training algorithms used for deep learning (such as SGD and its variants) can naturally handle soft labels (i.e., probability distributions over the label space); there is thus no need to force hard labels by sampling from the posterior. Second, by applying additive noise, we can repeatedly recompute the posterior distribution as better priors become available *without* consuming privacy budget. By following the "once and done" approach to label perturbation, we forgo the need to split training into disjoint stages.

To summarize, Ghazi et al. fix the prior at the beginning of the stage, choose parameters of the randomized response algorithm on the basis of this prior, and update the model by feeding it the perturbed label. We flip the order of these operations by applying a DP mechanism (additive Laplace) first, and then repeatedly use Bayesian inference by combining the prior that changes with each model update and the observables that do not.

To our knowledge, we are the first to explore connections between PATE and Label DP.

4 PATE with Semi-Supervised Learning

The PATE framework [28] splits the learning procedure into two stages. First, T teachers are trained on disjoint partitions of the private dataset. Second, a student acquires labels for training its own model by querying the teacher ensemble on samples drawn from a public (non-sensitive) unlabeled dataset. Differential privacy is enforced at the interface between the student and the teachers, by means of a private aggregation of the teachers' votes.

The critical observation underlying the PATE privacy analysis is that a private sample taints at most one teacher due to disjointedness of the teachers' training sets. The voting mechanism bounds the maximal impact of a single teacher's votes across all student queries by injecting noise in each voting instance. (We use the Confident Gaussian aggregation mechanism and its analysis from [29].)

The Label DP setting allows for unrestricted use of the features. We leverage this capability for training both teachers and the student:

- The unlabeled examples on which the student queries the teacher ensemble are random samples from the dataset *without* labels.
- All models—the teachers and the student—are trained with the FixMatch algorithm that uses (few) labeled examples and has access to the entirety of the original dataset as the supplementary unlabeled data.

Algorithms 1 and 2 adapt PATE to the Label DP setting. To draw a distinction between the PATE framework, which is generic and black-box, and its instantiation with the FixMatch algorithm for semi-supervised learning, we refer to the latter as PATE-FM.

The PATE framework does not require the student and the teacher models to have identical architecture, or share hyperparameters. In our experience, however, the optimal privacy/accuracy trade-offs force very similar choices on these models.

The selection of the number of teachers, T, balances two competing objectives. On the one hand, each teacher has access to n/T labeled and n unlabeled examples, which means that a smaller T corresponds to higher accuracy teacher models (due to more labeled examples per teacher). On the other hand, a larger T allows for a higher level of noise and lower privacy budget per query.

A similar dynamic controls K, the number of labels requested by the student. A higher number allows for more accurate training but inflates the privacy budget.

Algorithm 1: `PATE-FM.TrainTeacherEnsemble`

Input: Dataset $D \leftarrow \{(x_1, y_1), \ldots, (x_n, y_n)\}$, number of teachers T, training procedure `FixMatch`$(labeled\_data, unlabeled\_data)$

Define the unlabeled part of D: $D^- \leftarrow \{x_1, \ldots, x_n\}$
Partition D into T equal-sized disjoint subsets $D^{(1)}, \ldots, D^{(T)}$
for $i \leftarrow 1$ **to** T **do**
  $teacher_i \leftarrow$ `FixMatch`$(D^{(i)}, D^-)$
end
Output: Model ensemble $\{teacher_1, \ldots, teacher_T\}$

Algorithm 2: `PATE-FM.TrainStudent`

Input: Dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, number of label classes C, number of teachers T, training procedure `FixMatch`$(labeled\_data, unlabeled\_data)$, number of student samples K, noise parameters σ_1, σ_2, voting threshold τ

Define the unlabeled part of D: $D^- \leftarrow \{x_1, \ldots, x_n\}$
Initialize student dataset $D_S \leftarrow \emptyset$
$\{teacher_1, \ldots, teacher_T\} \leftarrow$ `TrainTeacherEnsemble`(D, T)
while $|D_S| < K$ **do**
  $x \leftarrow$ random sample from D^-
  for $i \leftarrow 1$ **to** C **do** // tallying up the votes
    $v_i \leftarrow |\{j : teacher_j(x) = i\}|$
  end
  if $\max_i\{v_i + \mathcal{N}(0, \sigma_1^2)\} \geq \tau$ **then**
    $y \leftarrow \operatorname{argmax}_i\{v_i + \mathcal{N}(0, \sigma_2^2)\}$
    $D_S \leftarrow D_S \cup \{(x, y)\}$
  end
end
$student \leftarrow$ `FixMatch`(D_S, D^-)
Output: Model $student$

Boosting teachers' accuracy for a given number of labels and minimizing the number of student queries and their privacy costs are key to producing high utility models with strong privacy guarantees. To this end, we use the state-of-the-art algorithm for semi-supervised learning FixMatch [32].

The FixMatch algorithm's primary domain is image classification: in addition to labelled and unlabelled examples, it assumes access to a pair of weak and strong data augmentation algorithms. The loss function of FixMatch is a weighted sum of two losses that enforce consistency of the model's prediction (i) between labeled examples and their weakly augmented versions, and (ii) between weakly and strongly augmented unlabeled images (restricted to high-confidence instances).

5 ALIBI: Additive Laplace with Iterative Bayesian Inference

We describe the general approach of Soft Randomized Response with post-processing and its concrete instantiation, ALIBI.

5.1 Soft Randomized Response

Consider the application of *Randomized Response (RR)* to the setting of Label DP. In its standard form, the random perturbation maps labels to labels, i.e., the label either retains its value with probability p or assumes a random value with probability $1 - p$.

We deviate from RR by replacing label randomization with a differentially private mechanism applied to a one-hot encoding of the training sample label; we refer to this mechanism as Soft Randomized Response (Soft-RR). By retaining more information (without significantly impacting privacy analysis), we support several possible post-processing algorithms (Section 5.2). Additionally, we argue that Soft-RR performs better than RR in practice since it does not force wrong hard labels.

Algorithm 3: Additive Laplace with Iterative Bayesian Inference, `ALIBI`

Input: Dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, noise parameter λ, Bayesian post-processing mechanism $\texttt{BPP}(\mathbf{o}, \lambda, prior)$ defined in Eq. (3), optimizer $\texttt{Update}(model, input, soft\_labels)$

Initialize noised dataset $D^* \leftarrow \emptyset$
for $i \leftarrow 1$ **to** n **do** `// noising labels`
  $\mathbf{o}_i \leftarrow \text{OneHot}(y_i) + \text{Laplace}(\lambda)$
  $D^* \leftarrow D^* \cup \{(x_i, \mathbf{o}_i)\}$
end

Randomly initialize model M
repeat
  for $(x, \mathbf{o})$ **in** D^* **do**
    $\mathbf{pred} \leftarrow M(x)$
    $\mathbf{post} \leftarrow \texttt{BPP}(\mathbf{o}, \lambda, \mathbf{pred})$
    $M \leftarrow \texttt{Update}(M, x, \mathbf{post})$
  end
until *convergence*;

Output: Model M

Following standard analysis (Dwork and Roth [14, Chapter 3.3]), ε-DP can be achieved with additive Laplace noise with per-coordinate standard deviation $2\sqrt{2}/\varepsilon$. Analogously, Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 \cdot \mathbb{I}_C)$ results in (ε, δ)-DP if $\sigma = \sqrt{2\ln(1.25/\delta)}/\varepsilon$ and $\varepsilon, \delta < 1$ ([14, Appendix A]).

5.2 Post-processing of Soft Randomized Response

The output of Soft-RR is no longer a valid soft label vector—values are not necessarily constrained to the [0,1] interval and do not sum up to 1. Thankfully, the fundamental property of DP allows privacy preservation under post-processing, which we use to obtain a probability mass function on perturbed labels to de-noise the mechanism's output. Treating the noisy labels $\mathbf{o}$ as the observables, and λ as the noise parameter, we can calculate the posterior $p(y = c \mid \mathbf{o}, \lambda)$ by Bayes' rule as:

$$p(y = c \mid \mathbf{o}, \lambda) = \frac{p(\mathbf{o} \mid y = c, \lambda) \cdot p(y = c)}{\sum_k p(\mathbf{o} \mid y = k, \lambda) \cdot p(y = k)}. \tag{1}$$

In the context of SGD, the current model's output can be interpreted as a prior probability distribution over the classes ($p(y)$).

ALIBI, or Additive Laplace with Iterative Bayesian Inference, is an algorithm that applies Bayesian post-processing to the output of the Laplace mechanism (Algorithm 3). With Laplace noise, the output of the mechanism for a specific label k is distributed as:

$$p(\mathbf{o} \mid y = k, \lambda) \propto e^{-|\mathbf{o}_k - 1|/\lambda} \prod_{j \neq k} e^{-|\mathbf{o}_j|/\lambda}. \tag{2}$$

Plugging (2) into (1) we derive:

$$p(y = c \mid \mathbf{o}, \lambda) = \text{SoftMax}(f(o_c)/\lambda + \log p(y = c)), \tag{3}$$

where $f(o_c) = -\sum_k |\mathbf{o}_k - [c = k]|$ and $[\cdot]$ is the Iverson bracket.

At a given privacy level, ALIBI empirically outperforms iterative Bayesian inference on additive Gaussian noise and naïve "uninformed" post-processing strategies, the details of which are deferred to Section D (Supplemental materials).

6 Evaluation

The algorithms are evaluated by training the Wide-ResNet architecture [38] on the CIFAR-10 and CIFAR-100 datasets [22]. The widening factor is set to 4 and 8 for CIFAR-10 and CIFAR-100 respectively. To facilitate comparison of privacy assured by the two approaches, we train our models

to achieve similar accuracy levels and provide the computed theoretical (ε, δ) upper bound. (Ignoring the privacy costs of hyperparameter tuning and of reporting the test set performance. The former can be minimized, with a modest loss in accuracy, by applying the selection procedure of Liu and Talwar [24].) We also provide the empirical privacy loss lower bound, ε_m, as estimated by the black-box attacks (Section 7).

PyTorch-based implementations of ALIBI, PATE-FM, and memorization attacks, including configuration options and hyperparameters, are publicly available [1].

6.1 PATE-FM

PATE-FM parameters are given in Table 3 (Supplemental materials). The student and the teacher models share the same architecture and are trained using FixMatch [32, 21] with hyperparameters from Sohn et al. [32] (we use the RandAugment variant [9]). The number of epochs is set so that the test accuracy reaches a plateau: for CIFAR-10, we train teachers for 40 epochs and the student for 200 epochs; for CIFAR-100, we train both the teachers and the student for 125 epochs. To report privacy, we set $\delta = 10^{-5}$. The student's accuracy is reported as the average of three runs (for a fixed teacher ensemble).

There are several hyperparameters affecting privacy/utility trade-off in PATE: number of teachers, level of noise, voting threshold, and number of labeled samples in the student dataset. Apart from the expected dependencies (increasing noise and the number of teachers benefits privacy at some accuracy cost), we observe the following:

- The number of labeled student samples beyond a certain level (10–25 samples per class) has little effect on the eventual model accuracy. This is in contrast with the standard (non-private) FixMatch where increasing the number of labeled samples usually improves accuracy.
- The voting threshold τ needs to be kept between 10% and 75% of the overall number of teachers. As pointed out in Papernot et al. [29], a lack of consensus between teachers hurts both privacy and utility, as the teacher ensemble errs more often on such inputs and a single vote can have more impact on the outcome. Setting the threshold too high leads to a selection bias in training samples, forcing the student model to train only on the most trivial samples. Even in the zero-noise regime, the threshold above 75% considerably hurts the accuracy.

6.2 ALIBI

We use mini-batch SGD with a batch size of 128 and a momentum of 0.9. We train all models for 200 epochs and pick the best checkpoint. We tune the learning rate, weight decay and noise multiplier to obtain the desired accuracy while minimizing privacy ε (see Table 4 in Supplemental materials). Since ALIBI is based on the Laplace mechanism, it achieves pure DP corresponding to $\delta = 0$.

6.3 Results

Table 1 presents the privacy budget and the memorization metric at three levels of accuracy across our two approaches for the both datasets. Additionally, for ALIBI, we provide the results of a higher accuracy model on the CIFAR-100 task; this higher accuracy was not attainable with PATE-FM.

For correct comparison with LP-2ST on CIFAR-100, we factor out modelling differences by applying both mechanisms to training the same architecture—ResNet18 [19, 25] with appropriate modifications to suit the task. Results are presented in Table 2. For reference, non-private baselines with this model achieve $\approx 95.5\%$ on CIFAR-10 and $\approx 79\%$ on CIFAR-100.

We note that there is a significant difference in computational resources required to train with PATE-FM and ALIBI. The reasons are twofold. First, in our experiments, models trained in a semi-supervised regime with few labeled examples take longer to converge. Second, PATE requires training the teacher ensemble, in which computational costs scale linearly with the size of the ensemble. Overall, this accounts for PATE-FM consuming 1,000–1,500$\times$ more computational resources (GPU·hr) compared to ALIBI.

Table 1: Privacy of PATE-FM [28, 32] and ALIBI, on CIFAR-10 and CIFAR-100 using Wide-ResNet18, matched by test accuracy levels. ε for PATE is at $\delta = 10^{-5}$. Empirical privacy loss ε_m is reported as a 95% confidence interval (CI).

Dataset	Accuracy level	Algorithm	Accuracy	ε	95%-CI ε_m
CIFAR-10	High	PATE-FM	93.7%	1.6	0.0–0.9
		ALIBI	94.0%	8.0	2.9–4.0
	Medium	PATE-FM	86.9%	0.29	0.0–0.6
		ALIBI	84.2%	2.1	1.0–2.2
	Low	PATE-FM	73.4%	0.18	0.0–0.4
		ALIBI	71.0%	1.0	0.0–2.2
CIFAR-100	Very High	PATE-FM	–	–	–
		ALIBI	75.3%	8.1	3.5–4.5
	High	PATE-FM	69.9%	715	0.4–1.0
		ALIBI	71.4%	6.3	2.8–3.5
	Medium	PATE-FM	50.0%	16	0.2–0.7
		ALIBI	51.6%	3.0	1.4–2.4
	Low	PATE-FM	30.5%	7.9	0.2–0.6
		ALIBI	31.4%	2.0	0.6–1.0

Table 2: Test accuracy of ALIBI and LP-2ST [18, Table 4], on CIFAR-100 applied to ResNet18, matched by ε.

Algorithm	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 5$	$\varepsilon = 6$	$\varepsilon = 8$
ALIBI	**55.0**	**65.0**	**69.0**	**71.5**	**74.4**
LP-2ST	28.7	50.2	63.5	70.6	74.1

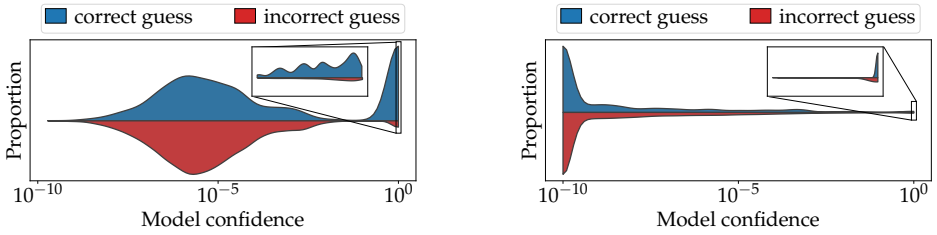
7 Measuring Memorization

We complement our evaluation with an empirical study of the (label) memorization abilities of classifiers trained with PATE-FM and ALIBI. While we can derive *provable* upper-bounds on the level of ε-Label DP provided by both algorithms, a direct comparison between the (actual) privacy of both algorithms is challenging as they rely on very different privacy analyses. As we will see, for CIFAR-100, PATE-FM empirically appears to provide much stronger privacy guarantees than ALIBI, even though the privacy analysis suggests the opposite.

To empirically measure privacy loss, we effectively instantiate the implicit adversary in the DP definition [26, 20]. We construct two datasets D, D' that differ only in the label of one example, and train a model on one of these two datasets. We then build an attack that "guesses" which of the two datasets was used. The attacker's advantage over a random guess can be used to derive an empirical lower bound on the mechanism's differential privacy. A formal treatment of memorization attacks using the framework of security games appears in Section B (Supplemental materials).

We point out the following differences between our approach and prior work. When providing lower bounds on privacy leakage, it is appropriate to consider the most powerful attacker pursuing the least challenging goal. In the context of standard DP (privacy for labels and features), it means an attacker that can add or remove a sample and seeks to determine whether the sample was part of the training dataset (i.e., membership inference [31]). In contrast, we operate within the constraints of Label DP, where the attacker's power is limited to manipulating a single label and the goal is to infer which label was used during training. (The closest analogue is attribute inference [37].)

Prior work replicates a memorization experiment many times (e.g., $1{,}000$ times [26]) to lower bound ε with standard statistical tools [26, 20]. As this approach is prohibitively expensive in our setting (prior work computed DP lower bounds for simpler datasets such as MNIST), we introduce the following heuristic: we train a *single* model on a training set with $1{,}000$ labels randomly flipped to a different class. For each of these "canaries" [6] with (flipped) label y', the attacker has to guess whether



(a) ALIBI (CIFAR-100, High)

(b) PATE-FM (CIFAR-100, High)

Figure 1: Comparison of the adversary's success rates in guessing the label of inserted canaries with ALIBI and PATE-FM (for CIFAR-100, with High accuracy level). We sort canaries by the model's maximal confidence on the two labels y', y'' that the adversary has to guess from. The violin plots show the proportion of correct guesses (the model is more confident in the canary label) and incorrect guesses. To maximize the adversary's distinguishing power, we limit the adversary to only issuing guesses for canaries where the model's confidence in either y' or y'' is above 99% (zoomed-in box). ALIBI is clearly more vulnerable to the attack (i.e., it is easier for the adversary to guess a canary's label), even though its theoretical privacy analysis yields a much lower bound on ε than for PATE-FM.

the canary's assigned label is y' or y'', where $y'' \neq y'$ is a different random incorrect label. If the adversary has accuracy α in this guessing game, we can lower bound ε by $1 - \log(\frac{\alpha}{1-\alpha})$. As in prior work [26, 20], we compute a 95% confidence interval on the adversary's guessing accuracy α and thus on the level of privacy ε against our attack. We formally define our attacking game and analyze the relationship between the attacker's success and the privacy budget ε in Section B (Supplemental materials).

To isolate the impact that can be attributed to the difference between privacy-preserving training procedures, we apply ALIBI and PATE-FM to the same target model architecture and tune their privacy parameters to achieve similar accuracy levels. (For discussion of confounders, see Erlingsson et al. [15].) Furthermore, we deliberately limit memorization attacks to be trainer-agnostic, to avoid privileging, for instance, a more complex training procedure over a simpler one.

Our empirical estimates of the privacy loss ε_m are given in Table 1. The estimates for ALIBI are approximately within a factor of 2 of the theoretical upper bound on ε. This suggests that the theoretical privacy analysis of ALIBI is close to tight. On the other hand, for PATE-FM our empirical estimates of the privacy loss are up to three orders of magnitude smaller than the theoretical upper bound. While we have no guarantee that our attack is the strongest possible (and it likely is not), our experiment does suggest that for a fixed accuracy level, PATE-FM is much less likely to memorize labels compared to ALIBI (see Figure 1). Empirically, it thus appears that PATE-FM provides a much better privacy-utility tradeoff than ALIBI, even though the theoretical analysis suggests the opposite.

This being said, we encourage critical interpretation of privacy claims backed by memorization estimates. Attacks, and the corresponding lower bounds on privacy loss, are provisional and conditional; they can be shattered with better algorithms or additional resources. The empirical lower bounds should be contrasted with provable upper bounds that provide the most conservative guidance in selecting privacy-preserving algorithms and setting their parameters.

8 Conclusion

We have proposed and evaluated two approaches toward achieving Label DP in ML: ALIBI and PATE-FM (the PATE framework instantiated with FixMatch). We have demonstrated that both approaches can achieve state-of-the-art results on standard datasets. As the two approaches have different theoretical foundations and exhibit very different performance characteristics, they also present an interesting combination of trade-offs.

PATE-FM offers impressive empirical privacy despite a very conservative theoretical privacy analysis. On the other hand, its training setup is fairly complicated. The dependency on semi-supervised learning techniques further restricts the tasks this method can be used for at present.

In contrast, ALIBI's implementation and interpretation are quite simple. (1) It is a generic algorithm that is compatible with any iterative procedure for ML training, such as SGD or its variants. It is not a

black-box however, and requires the ability to update training data labels as they are processed by the training algorithm. (2) It enjoys a tighter privacy analysis resulting in a more realistic upper bound on the privacy budget, but appears to offer less privacy than PATE-FM empirically. (3) It can be used to train models to accuracy levels which PATE cannot achieve without significantly more data.

In addition to privacy and performance trade-offs, our two algorithms also differ significantly in their time and computational resource requirements (as discussed in the Section 6.3).

Beyond the concrete improvements offered by our new techniques, our work emphasizes the need for more research across multiple avenues: (1) bridging the gap between privacy lower bounds, backed by attacks, and upper bounds, based on privacy analyses (in particular for PATE); (2) making the privacy analysis of different approaches comparable; (3) designing stronger attack models to better capture empirical privacy guarantees.

9 Acknowledgements and Funding Sources

The authors are grateful to NeurIPS anonymous reviewers and area chairs for their helpful comments. Florian Tramèr is supported by NSF award CNS-1804222.

References

[1] `https://github.com/facebookresearch/label_dp_antipodes`.

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. "A closer look at memorization in deep networks". In: *International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 233–242.

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. "Learning with pseudo-ensembles". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. 2014.

[4] Amos Beimel, Kobbi Nissim, and Uri Stemmer. "Private learning and sanitization: pure vs. approximate differential privacy". In: *Theory of Computing* 12.1 (2016), pp. 1–61.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. "MixMatch: A holistic approach to semi-supervised learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. 2019.

[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. "The Secret Sharer: Evaluating and testing unintended memorization in neural networks". In: *28th USENIX Security Symposium*. 2019, pp. 267–284.

[7] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. "Extracting training data from large language models". In: *30th USENIX Security Symposium*. 2021, pp. 2633–2650.

[8] Kamalika Chaudhuri and Daniel Hsu. "Sample Complexity Bounds for Differentially Private Learning". In: *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*. Ed. by Sham M. Kakade and Ulrike von Luxburg. Vol. 19. June 2011, pp. 155–186.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. "RandAugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.

[10] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. "Efficient Projections onto the ℓ_1-ball for learning in high dimensions". In: *Proceedings of the 25th International Conference on Machine Learning (ICML)*. 2008, pp. 272–279.

[11] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. "Preserving statistical validity in adaptive data analysis". In: *Proceedings of the forty-seventh annual ACM Symposium on Theory of Computing (STOC)*. 2015, pp. 117–126.

[12] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. "Our data, ourselves: Privacy via distributed noise generation". In: *Advances in Cryptology—EUROCRYPT 2006*. Ed. by Serge Vaudenay. 2006, pp. 486–503.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating noise to sensitivity in private data analysis". In: *Proceedings of the 3rd Conference on Theory of Cryptography (TCC)*. Ed. by Shai Halevi and Tal Rabin. 2006, pp. 265–284.

[14] Cynthia Dwork and Aaron Roth. "The algorithmic foundations of differential privacy." In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.

[15] Úlfar Erlingsson, Ilya Mironov, Ananth Raghunathan, and Shuang Song. "That which we call private". In: *CoRR* abs/1908.03566 (2019).

[16] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. "RAPPOR: Randomized aggregatable privacy-preserving ordinal response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Nov. 2014, pp. 1054–1067.

[17] Vitaly Feldman. "Does learning require memorization? A short tale about a long tail". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. 2020, pp. 954–959.

[18] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. *On deep learning with label differential privacy*. 2021. arXiv: 2102.06062 [cs.LG].

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. "Auditing differentially private machine learning: How private is private SGD?" In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. 2020, pp. 22205–22216.

[21] Jungdae Kim. *FixMatch-pytorch*. https://github.com/kekmodel/FixMatch-pytorch. Available under MIT License.

[22] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[23] Dong-Hyun Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. 2013.

[24] Jingcheng Liu and Kunal Talwar. "Private Selection from Private Candidates". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. 2019, pp. 298–309.

[25] Kuang Liu. https://github.com/kuangliu/pytorch-cifar/blob/master/models/resnet.py. Available under MIT License.

[26] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. "Adversary instantiation: Lower bounds for differentially private machine learning". In: *2021 IEEE Symposium on Security and Privacy (S&P)*. May 2021, pp. 1183–1199.

[27] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. "The geometry of differential privacy: The sparse and approximate cases". In: *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing (STOC)*. 2013, pp. 351–360.

[28] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. "Semi-supervised knowledge transfer for deep learning from private training data". In: *5th International Conference on Learning Representations (ICLR)*. 2017.

[29] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. "Scalable private learning with PATE". In: *6th International Conference on Learning Representations (ICLR)*. Code available from https://github.com/tensorflow/privacy/tree/master/research/pate_2018 under Apache 2.0 license. 2018.

[30] Paul M Schwartz. "Privacy and democracy in cyberspace". In: *Vanderbilt Law Review* 52 (1999), pp. 1607–1702.

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models". In: *2017 IEEE Symposium on Security and Privacy (S&P)*. 2017, pp. 3–18.

[32] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. "FixMatch: Simplifying semi-supervised learning with consistency and confidence". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. 2020, pp. 596–608.

[33] Salil Vadhan. "The complexity of differential privacy". In: *Tutorials on the Foundations of Cryptography*. Springer, 2017, pp. 347–450.

[34] Di Wang and Jinhui Xu. "On sparse linear regression in the local differential privacy model". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. PMLR, June 2019, pp. 6628–6637.

[35] Weiran Wang and Miguel Á. Carreira-Perpiñán. *Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application*. 2013. arXiv: 1309.1541 [cs.LG].

[36] Stanley L Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.

[37] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. "Privacy risk in machine learning: Analyzing the connection to overfitting". In: *31st IEEE Computer Security Foundations Symposium (CSF)*. 2018, pp. 268–282.

[38] Sergey Zagoruyko and Nikos Komodakis. "Wide residual networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith. Sept. 2016, pp. 87.1–87.12.

[39] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *5th International Conference on Learning Representations (ICLR)*. 2017.

A Hyperparameters

Tables 3 and 4 summarize hyperparameters for PATE-FM and ALIBI respectively.

Table 3: PATE-FM (Algorithms 1 and 2) hyperparameters for select accuracy levels.

Dataset	Teachers	σ_1	σ_2	τ	Queries answered	Accuracy Ensemble	Accuracy Student
CIFAR-10	200	160	20	100	500	90.8%	93.7%
	800	800	300	400	250	60.8%	86.9%
	800	800	500	400	250	36.4%	73.4%
CIFAR-100	20	7	2	2	9,400	74.0%	69.9%
	100	45	15	10	1,000	55.0%	50.0%
	100	90	30	10	1,000	28.8%	30.5%

Table 4: Hyperparameters of ALIBI (Algorithm 3) for select accuracy levels.

Dataset	Learning rate	Weight decay ($\times 10^{-4}$)	Accuracy
CIFAR-10	0.308	0.568	94.0%
	0.960	0.00796	84.2%
	0.315	1.50	71.0%
CIFAR-100	0.0037	33.5	75.3%
	0.0057	17.5	71.4%
	0.100	2.33	51.6%
	0.1	1.00	31.4%

B Memorization: Formal treatment

To empirically bound the level ε of DP, prior work instantiates a general *membership inference* game, defined in Figure 2 for two arbitrary neighboring datasets D_0 and D_1.

By repeating this game multiple times, we can estimate the adversary's success rate and convert this into a lower bound on ε.

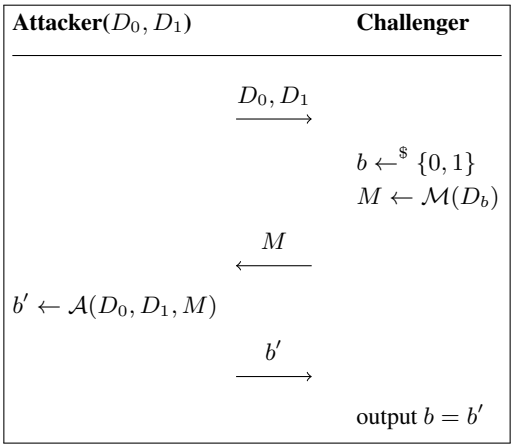


Figure 2: Basic membership inference game, Game 1.

This would be prohibitively expensive in our setting (each iteration of the game requires training a model on CIFAR-10 or CIFAR-100, and the game has to be repeated about 1,000 times to get non-trivial bounds). We thus propose a heuristic approach for running multiple iterations of this game while training a *single* model.

First, we will change the game slightly so as to allow the adversary to *abstain* from issuing a guess on some instances. That is, the output range of $\mathcal{A}$ is $\{0, 1, \perp\}$. We then define the adversary's *correct guess rate* (CGR) as:

$$\mathrm{CGR}_{D_0,D_1} \coloneqq \Pr[b = b' \mid b' = \mathcal{A}(D_0, D_1, \mathcal{M}(D_b)) \wedge b' \neq \perp] \ .$$

The probability is taken over the bit b, the randomness of the mechanism $\mathcal{M}$ and the algorithm $\mathcal{A}$.

Theorem B.1. *If $\mathcal{M}$ satisfies ε-DP and D_0, D_1 are two adjacent databases, then*

$$\varepsilon \geq \log\left(\frac{\mathrm{CGR}_{D_0,D_1}}{1 - \mathrm{CGR}_{D_0,D_1}}\right).$$

Proof. For notational convenience, define A_x as a random variable distributed according to $\mathcal{A}(D_0, D_1, \mathcal{M}(D_x))$ for $x \in \{0, 1, b\}$. Then

$$\begin{aligned}
\frac{\mathrm{CGR}_{D_0,D_1}}{1 - \mathrm{CGR}_{D_0,D_1}} &= \frac{\Pr[b' = b \mid b' = A_b \wedge b' \neq \perp]}{\Pr[b' = 1 - b \mid b' = A_b \wedge b' \neq \perp]} \\
&= \frac{\Pr[A_b = b]}{\Pr[A_b = 1 - b]} \\
&= \frac{\Pr[A_0 = 0] + \Pr[A_1 = 1]}{\Pr[A_0 = 1] + \Pr[A_1 = 0]} \\
&\leq \max\left\{\frac{\Pr[A_0 = 0]}{\Pr[A_1 = 0]}, \frac{\Pr[A_0 = 1]}{\Pr[A_1 = 1]}\right\} \qquad \text{(by the mediant inequality)} \\
&\leq e^{\varepsilon} \ ,
\end{aligned}$$

where the last inequality follows from the assumption that $\mathcal{M}$ is ε-DP. □

It now remains to be seen how we can bound the adversary's correct guessing rate CGR. We define Game 2 (Figure 3) by making a small change to Game 1 above, so that the neighboring datasets D_0 and D_1 are chosen *at random* in each iteration of the game, by flipping the label of one example of a common dataset D.

Similar to Game 1, we define the adversary's probability of winning in Game 2 conditional on $\mathcal{A}$'s output not being $\perp$. Let the *average* CGR in Game 2 (ACGR) be:

$$\begin{aligned}
\mathrm{ACGR}_D &\coloneqq \mathop{\mathbb{E}}_{D_0,D_1}\left[\Pr[b = b' \mid b' = \mathcal{A}(D_0, D_1, \mathcal{M}(D_b)) \wedge b' \neq \perp]\right] \\
&= \mathop{\mathbb{E}}_{D_0,D_1}\left[\mathrm{CGR}_{D_0,D_1}\right] \ .
\end{aligned}$$

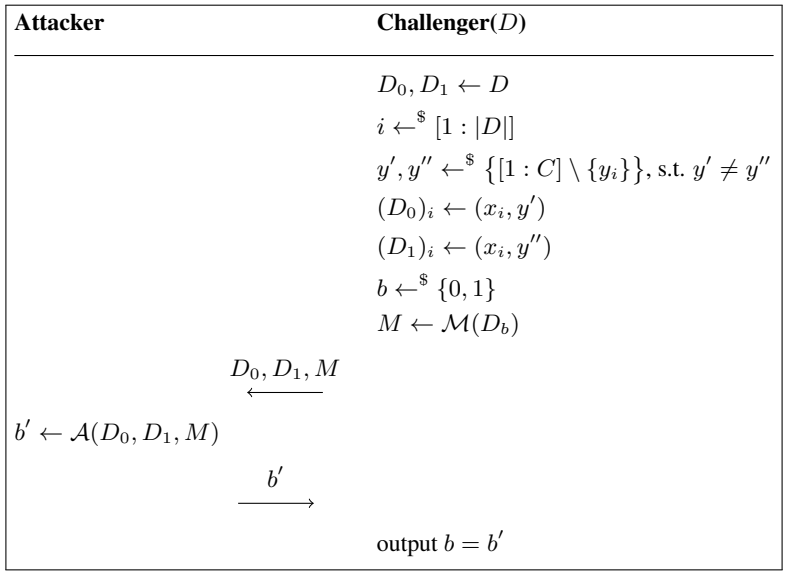


Figure 3: D_0 and D_1 are defined randomly (Game 2).

If we can lower-bound ACGR by some value α, then there exists at least one pair of neighboring datasets D_0, D_1 such that $\mathrm{CGR}_{D_0,D1} \geq \alpha$. As the DP guarantee has to hold for *all* neighboring datasets, we can use a bound on ACGR to bound ε.

Finally, instead of repeating Game 2 many times to get a bound on ACGR, we instead simulate multiple iterations of Game 2 at once, which becomes Game 3 (formally defined in Figure 4). The heuristic step here is that we assume that each of the N guesses made by the adversary are independent from each other and reflect the adversary's guesses in N independent iterations of Game 2.

Given one iteration of Game 3 with N "canaries", we can compute a lower bound on the adversary's average CGR using standard confidence intervals for a Binomial random variable. That is, we count the number of correct guesses among the $M \leq N$ instances where the adversary made a guess $b'_i \neq \perp$, and apply a Clopper-Pearson bound.

We can improve the tightness of this bound further. In Game 3, for each canary (x_i, y'_i) that the model is trained on (assuming $b_i = 0$), we record the adversary's guess with respect to only one other random label y''. Yet, we could record the adversary's guess with respect to *all* $C - 2$ choices of $y''_i \neq y_i, y'_i$ to get a tighter estimate of the adversary's average success rate. However, we definitely cannot treat these guesses as independent. Instead, we first estimate the adversary's (empirical) average correct guessing rate $\overline{\mathrm{ACGR}}_i$ for each canary (where the average is taken over all possible choices for y''_i):

$$
\begin{aligned}
b'_{i,j} &\coloneqq \mathcal{A}(D^{(i,j)}_{b_{i,j}}, D^{(i,j)}_{1-b_{i,j}}, M) \quad \text{for } j = 1, \ldots, C-2, \\
m_i &\coloneqq \sum_{j=1}^{C-2} [b'_{i,j} \neq \perp], \\
\overline{\mathrm{ACGR}}_i &\coloneqq \frac{1}{m_i} \sum_{j=1}^{C-2} [b'_{i,j} = b_{i,j}],
\end{aligned}
$$

where $b_{i,j}$ are iid uniform bits, $D^{(i,j)}_0$ are D for all i and j, $D^{(i,j)}_1, \ldots, D^{(i,j)}_1$ are copies of D with the $C - 2$ possible choices for the label y''_i, and $[\cdot]$ is the Iverson bracket. (By convention, $0/0 = 0$.) We then compute a confidence interval for the empirical mean of all the $\overline{\mathrm{ACGR}}_i$. As the adversary may abstain from making a guess with a different probability for each canary (i.e., the m_i's may not all be equal) we have to weigh the $\overline{\mathrm{ACGR}}_i$ values accordingly. That is, we compute the *weighted* average and standard deviation of the $\overline{\mathrm{ACGR}}_i$ with the M_i as *reliability weights*. Finally, we obtain

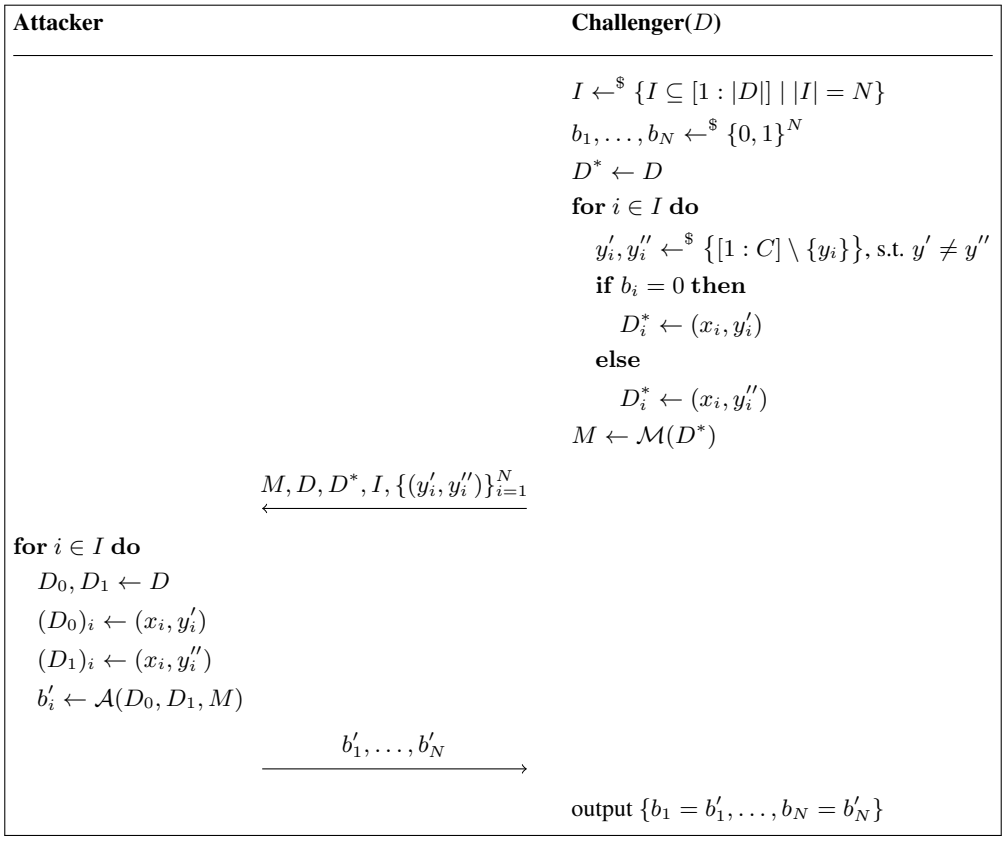


Figure 4: Game 3.

a confidence interval for the adversary's $\overline{\mathrm{ACGR}}$ using a standard 95% confidence interval for the normal distribution.

Finally, it remains to define our adversary $\mathcal{A}(D_0, D_1, M)$ where $(x_i, y') \in D_0$ and $(x_i, y'') \in D_1$. The adversary simply looks at the model's confidence on x_i for both possible labels and guesses that the more confident of the two is the label that the model was trained on. However, if both labels have confidence below some fixed threshold τ, the adversary abstains. Formally:

$$\mathcal{A}(D_0, D_1, M) = \begin{cases} \bot & \text{if } \max(M(x_i)_{y'}, M(x_i)_{y''}) < \tau \\ [M(x_i)_{y'} > M(x_i)_{y''}] & \text{otherwise} \end{cases}.$$

We consider different thresholds $\tau \in [0.5, 0.99]$ and report the best resulting attack (i.e., the setting with the highest lower-bound on $\overline{\mathrm{ACGR}}$). For simplicity, we omit corrections for multiple hypothesis testing.

C Memorization: Validating the Heuristic

The previous section introduces a sequence of security games (Figures 2–4) that relate the success probability of a membership inference adversary (Game 1) to an efficient computational procedure (Game 3). It starts by randomizing a single instance of Game 1 into Game 2. In order to compute the adversary's probability of winning Game 2 with sufficient accuracy, the experiment needs to be repeated hundreds of times. Doing so would be prohibitively expensive as each run requires training a new model from scratch.

We use a heuristic whereby the independent runs of Game 2 are replaced with correlated instances of the membership inference game that share the same trained model (Game 3). Concretely, it means that instead of introducing a single canary (a mislabeled input) into a training dataset, Game 3 injects

multiple canaries all at once. While doing so does change the input distribution of the training procedure, we argue that the overall effects are minimal and do not qualitatively affect our findings.

We test the heuristic's validity by comparing the adversary's advantage against ALIBI in Game 3, where the number of simultaneously inserted canaries is $N = 1{,}000$ as in Table 1, with ten repetitions of Game 3 with $N = 100$. The results are presented in Table 5. Notably, the 95% confidence intervals for ε_m are in very close agreement, thus supporting the heuristic.

Table 5: Comparison of 95% confidence intervals (CI) for the membership inference adversary against ALIBI given a single run of Game 3 with $N = 1000$ and 10 runs of Game 3 with $N = 100$.

		95%-CI ε_m	
Dataset	Accuracy level	1,000 labels	10×100 labels
CIFAR-10	High	2.9–4.0	2.7–3.6
	Medium	1.0–2.2	0.2–2.5
	Low	0.0–2.2	0.0–1.6
CIFAR-100	High	2.8–3.5	2.7–3.4
	Medium	1.4–2.4	1.3–2.0
	Low	0.6–1.0	0.6–1.1

D Post-processing for Soft Randomized Response

For completeness, we describe Soft RR variants instantiated with uninformed post-processing and the Gaussian mechanism. We found that ALIBI dominates alternatives by achieving better accuracy with stronger privacy. In particular, ALIBI's upper bounds are stronger than those of the Gaussian mechanisms for the same levels of accuracy, while their empirical privacy losses are statistically indistinguishable.

D.1 Uninformed post-processing

Given Soft-RR's output vector $\mathbf{o}$, we may reduce error by mapping it to the closest point on the probability simplex (compare with Nikolov et al. [27]). In other words, we are solving the following constrained optimization problem:

$$\min \|\mathbf{o}^* - \mathbf{o}\|_2 \quad \text{subject to } \begin{cases} \forall i \;\; 0 \le o_i^* \le 1, \\ \sum_i o_i^* = 1 \end{cases}$$

The problem is (strictly) convex, and thus admits a unique, efficiently computable solution. Moreover, a particularly simple and efficient method (Algorithm 4) exists due to Duchi et al. [10] (see also Wang and Carreira-Perpiñán for other approaches [35]).

Algorithm 4: Post-processing using Min Projection (Duchi et al. [10]).

Input: $\mathbf{o} = (o_1, \ldots, o_C) \in \mathbb{R}^C$
Output: Projection of $\mathbf{o}$ onto the probability simplex
Sort $\mathbf{o}$ as $s_1 \ge s_2 \ge \cdots \ge s_C$
Find $k \leftarrow \max_j \left\{ j \in [1:C] \colon s_j > \frac{1}{j}(\sum_{i=1}^{j} s_i - 1) \right\}$
$u \leftarrow \frac{1}{k}(\sum_{i=1}^{k} s_i - 1)$
for $i \leftarrow 1$ **to** C **do**
  $o_i' \leftarrow \max(o_i - u, 0)$
end
Output $\mathbf{o}'$

D.2 Bayesian post-processing on Additive Gaussian Mechanism

This follows the same post-processing algorithm as ALIBI as described in Section 5.2, except for Gaussian noise instead of Laplace. Eq. (2) changes to the following (note the switch of the noise

parameter from λ to σ):

$$p(\mathbf{o} \mid y = k, \sigma) \propto e^{-\frac{(\mathbf{o}_k - 1)^2}{2\sigma^2}} \prod_{j \neq k} e^{-\frac{\mathbf{o}_j^2}{2\sigma^2}}. \tag{4}$$

Plugging (4) in (1) we have:

$$p(y = c \mid \mathbf{o}, \sigma) = \frac{e^{\mathbf{o}_c/\sigma^2} \cdot p(y = c)}{\sum_k e^{\mathbf{o}_k/\sigma^2} \cdot p(y = k)} = \mathrm{SoftMax}(\mathbf{o}_c/\sigma^2 + \log p(y = c)). \tag{5}$$

The training algorithm as described in Algorithm 3 can be modified by setting `BPP` to (5) to work in this setting.

The results of applying the Gaussian mechanism ("AGIBI") are reported in Table 6 together with ALIBI performance from Table 1 for ease of comparison. The claimed privacy losses (the ε column) strongly favor ALIBI over the Gaussian mechanism; the empirically computed privacy loss lower bounds do not separate the two mechanisms.

Table 6: ALIBI and AGIBI on CIFAR-10 and CIFAR-100 using Wide-ResNet18, matched by test accuracy levels. Empirical privacy loss ε_m is reported as a 95% confidence interval (CI).

Dataset	Accuracy level	Algorithm	Accuracy	ε	95%-CI ε_m
CIFAR-10	High	ALIBI	94.0%	8.0	2.9–4.0
		AGIBI	93.5%	19	2.1–3.2
	Medium	ALIBI	84.2%	2.1	1.0–2.2
		AGIBI	84.3%	7.7	0.7–1.4
	Low	ALIBI	71.0%	1.0	0.0–2.2
		AGIBI	71.3%	3.7	0.0–2.0
CIFAR-100	High	ALIBI	71.4%	6.3	2.8–3.5
		AGIBI	69.9%	17	2.7–3.4
	Medium	ALIBI	51.6%	3.0	1.4–2.4
		AGIBI	50.8%	8.4	1.3–2.0
	Low	ALIBI	31.4%	2.0	0.6–1.0
		AGIBI	28.7%	5.4	0.6–1.1