Auditing Private Prediction

Karan Chadha[1] **Matthew Jagielski**[2] **Nicolas Papernot**[2] **Christopher A. Choquette-Choo**[2] **Milad Nasr**[2]

Abstract

Differential privacy (DP) offers a theoretical upper bound on the potential privacy leakage of an algorithm, while empirical auditing establishes a practical lower bound. Auditing techniques exist for DP training algorithms. However machine learning can also be made private at inference. We propose the first framework for auditing private prediction where we instantiate adversaries with varying poisoning and query capabilities. This enables us to study the privacy leakage of four private prediction algorithms: PATE (Papernot et al., 2016), CaPC (Choquette-Choo et al., 2020), PromptPATE (Duan et al., 2023), and Private-kNN (Zhu et al., 2020). To conduct our audit, we introduce novel techniques to empirically evaluate privacy leakage in terms of Renyi DP. Our experiments show that (i) the privacy analysis of private prediction can be improved, (ii) algorithms which are easier to poison lead to much higher privacy leakage, and (iii) the privacy leakage is significantly lower for adversaries without query control than those with full control.

1. Introduction

Differential privacy (DP) assesses an algorithm's privacy by examining its outputs on two adjacent datasets, S and S', which differ in one data point (Dwork et al., 2006). It bounds the log ratio of output distribution probabilities on these datasets using a parameter ε. A small ε ensures that an adversary cannot confidently distinguish whether the algorithm processed S or S'. Thus, ε analytically bounds private information leakage from the algorithm's outputs. In contrast, *auditing* a private algorithm (Ding et al., 2018; Jagielski et al., 2020) provides a lower bound on its privacy leakage. Analyzing both upper and lower bounds can yield three insights: a large gap may indicate a potential slack in the analysis; a lower bound surpassing the upper may indicate an incorrect analysis or implementation (Tramer et al., 2022); and tracking how these bounds shift with changes to assumptions on the adversary's capabilities and knowledge can inform us of which assumptions contribute most to the algorithm's privacy leakage (Nasr et al., 2021).

In the context of machine learning (ML), existing work has exclusively audited differentially private *training* algorithms. These algorithms train models satisfying DP (Abadi et al., 2016), ensuring the privacy of all model predictions due to DP's post-processing property.[1] However, machine learning can also be made private at *inference*. Here, models are trained non-privately and their predictions are noised before release to satisfy DP. Despite the increasing relevance of private prediction, notably to task adaptation of large language models (Duan et al., 2023), there are no known techniques to audit such algorithms. Our work addresses this gap and proposes the first auditing framework to do so.

Private prediction algorithms diverge from private training by training multiple non-private teacher models on separate data partitions. At inference, they compile individual model predictions into a histogram, introduce noise to each histogram bin, and select the most frequent bin as the output. To quantify the privacy leakage of private prediction, we upper bound several cumulants of the log ratio of output distributions (instead of the log ratio directly), resulting in a Renyi DP (RDP) guarantee. RDP, an alternative formulation of DP, offers enhanced compositional properties which is beneficial when assessing privacy leakage that is composed across multiple test queries. When reporting, we convert the composed RDP guarantees to a classical DP guarantee.

To audit private prediction, the standard approach would consider the full output set of multiple queries and audit the resulting distribution using classical auditing techniques. However, the discrete high dimensional nature of the output distribution makes the application of these techniques non-trivial. Moreover, composition theorems in classical DP analysis are lossy and composing corresponding lower bounds is invalid. This necessitates developing a novel methodology to audit RDP guarantees which we use to

[1]Stanford University, Part of work done while an intern at Google DeepMind. [2]Google DeepMind. Correspondence to: Karan Chadha <knchadha@stanford.edu>, Matthew Jagielski <jagielski@google.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] An arbitrary function can be applied to the outputs of a DP algorithm with no consequences to privacy.

audit per-query privacy leakage and further compose using lossless RDP composition theorems to give lower bounds on the privacy leakage across multiple queries. Along with our new approach to RDP auditing, we also introduce a technique to calculate the exact Renyi divergence between the outputs of noisy argmax on neighboring histograms. This dual approach enables a more nuanced assessment and helps in attributing discrepancies between the audit and the theoretical analysis to either looseness in the analysis or strength of the adversary, thereby enhancing the clarity of our audit.

We apply our auditing framework to four well-known private prediction algorithms: PATE (Papernot et al., 2016; 2018), CaPC (Choquette-Choo et al., 2020), PromptPATE (Duan et al., 2023) and Private-kNN (Zhu et al., 2020). All of these algorithms except Private-kNN allow for a data-dependent privacy analysis, where the standard data-independent analysis can be refined in cases when most models agree on the prediction to be made. Even so, our exact privacy analysis is tighter than this data-dependent calculation indicating potential slack in the analysis. Across all algorithms, we find that an adversary's capability to control the test queries is an important factor contributing to maximal privacy leakage. Furthermore, more privacy is leaked in cases where it is easier for adversaries to impact the behavior of a teacher model on multiple test queries. These trends are highlighted not only by the leakage due to different adversaries, but also due to the differences in the relative leakage of different private prediction algorithms correlating with their ease of poisoning.

1.1. Related Work

Malek Esmaeili et al. (2021) run attacks on PATE under label DP (a variant of DP protecting just the labels), for a specific type of adversary. Our goal is to audit private prediction in general, and with more granularity to enable improved attacks or analysis. Wang et al. (2022) show *test-time* attacks on private prediction, leaking privacy of the queries made to private prediction algorithms, while our goal is to measure privacy leakage of the training data.

Ding et al. (2018) propose an approach to detect DP violations in many classic DP algorithms including noisy argmax, which we consider in our work. Our work requires RDP auditing rather than their ϵ-DP auditing, and we audit using inputs that are relevant to private prediction, to measure the privacy leakage from these ML algorithms rather than just the noisy argmax.

Our work measures leakage of private prediction algorithms through their returned labels. Choquette-Choo et al. (2021); Li & Zhang (2021) demonstrate membership inference attacks on traditional classifiers using only returned labels. Recent work on auditing has shown how to lower bound general RDP guarantees using variational characterizations (Kong et al., 2023); our work instead performs audits suited for mechanisms with a discrete output space and with an attack interpretation.

DP in Machine Learning DP-SGD (Abadi et al., 2016; Bassily et al., 2014) is the canonical method for performing machine learning while satisfying finite DP guarantees. There have been many improvements over the past decade, including to tighten the theoretical analysis leading to reduced ϵ parameters (Balle et al., 2018; Kairouz et al., 2021; Denisov et al., 2022; Choquette-Choo et al., 2022; 2023a;b) and other empirical tricks to improve the utility of trained models (De et al., 2022; Papernot et al., 2021). These improvements have enabled production deployments of DP ML models, for examples in GBoard on-device models (Xu et al., 2023). However, practical use cases of DP ML such as these require a large $\epsilon > 1$ to obtain reasonable utility. Auditing helps understand if this analysis is tight and which assumptions may be contributing most to the high privacy budget.

Privacy Testing/Auditing There exist several other recently proposed auditing methods for DP training. Nasr et al. (2023) show how to audit DP-SGD using only two training runs, which yields tight audits for natural datasets. Steinke et al. (2023) show how to perform auditing of DP-SGD in a single training run. However, their approach comes at the cost of requiring many injected "canary" examples to be simultaneously memorized. Our work performs the first audits of private predictions, and designing sets of canary examples for private prediction algorithms is interesting future work. Finally, Andrew et al. (2023) show how to estimate (not lower bound) the privacy parameter ϵ in a single training run and focus on a federated learning setting. Pillutla et al. (2024) audits a lifted notion of DP with multiple randomized canaries and builds adaptive confidence intervals to give better audits requiring fewer model training runs.

2. Notation and Preliminaries

We work in the multiclass classification setting where we use features from $\mathcal{X} \subset \mathbb{R}^d$ to predict labels from $\mathcal{Y} = 1, \ldots, C$. Let the space of feature vector and label pairs be $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, with $S = (x_i, y_i)_{i=1}^n$ the training dataset. In ML, our task is to learn a function (model) $f(\cdot; S) : \mathcal{X} \to \mathcal{Y}$ using the training dataset S, such that $f(x; S)$ accurately estimates the label y of the feature x. A long line of work has shown that machine learning models can leak sensitive information about its training dataset (Shokri et al., 2016; Tramèr et al., 2022; Carlini et al., 2021). The gold standard for preventing this is differential privacy.

2.1. Differential Privacy

Definition 1 (Approximate DP)**.** (Dwork et al., 2006) An algorithm $\mathcal{A} : \mathcal{S}^n \to \mathcal{O}$ is (ε, δ)-DP if for any measurable $O \in \mathcal{O}$ and $S, S' \in \mathcal{S}^n$ that differ in one entry, we have $\mathbb{P}\left(\mathcal{A}(S) \in O\right) \leq e^{\varepsilon} \cdot \mathbb{P}\left(\mathcal{A}(S') \in O\right) + \delta$.

An important property of DP is sequential composition: given two mechanisms $\mathcal{A}_1, \mathcal{A}_2$ each satisfying a finite DP bound, $\mathcal{A}_2(S, \mathcal{A}_1(S))$ also has a finite DP bound. However, the composition bounds for approximate DP are not exact and lead to loose privacy accounting. Renyi DP (RDP) is an alternative definition to (ε, δ)-DP that provides lossless composition bounds. Thus, RDP is the analysis method of choice for private prediction algorithms. The Renyi divergence between distributions P and Q is defined as:

$$D_\alpha(P||Q) := \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim P} \left\{ \frac{P(x)}{Q(x)}^{\alpha} \right\}.$$

Definition 2 (Renyi DP)**.** (Mironov, 2017) An algorithm $\mathcal{A} : \mathcal{S}^n \to \mathcal{O}$ satisfies $(\alpha, \varepsilon_\alpha)$-RDP, if for any $S, S' \in \mathcal{S}^n$ that differ in one entry we have $D_\alpha(\mathcal{A}(S)||\mathcal{A}(S')) \leq \varepsilon_\alpha$.

The RDP guarantee for the composition of mechanisms is simply the order-wise sum of individual RDP guarantee. We keep track of the RDP for several orders, and report the optimal (ε, δ)-DP guarantee across orders found using the following theorem.

Theorem 1. *(Balle et al., 2020, Thm. 20) If an algorithm is $(\alpha, \varepsilon_\alpha)$-RDP, then it is also $(\varepsilon_\alpha + \log(\frac{\alpha-1}{\alpha}) - \frac{\log\delta + \log\alpha}{\alpha - 1}, \delta)$-DP for any $\delta \in (0, 1)$.*

2.2. Auditing Differential Privacy

Unlike the upper bounds that any DP algorithm must satisfy, auditing gives a lower bound. We define *auditing* the approximate DP guarantees of an algorithm $\mathcal{A}$ as empirically estimating parameters ε^ℓ and δ such that the algorithm does not satisfy (ε, δ)-DP for any $\varepsilon < \varepsilon^\ell$. We denote it as ε^ℓ since it is a lower bound on the true (ε, δ)-DP satisfied by the algorithm. To find such a lower bound for an algorithm $\mathcal{A}$, we choose neighbouring datasets S, S' and an output set O on which we run the algorithm $\mathcal{A}$ on S and S' many times (say T) and check the number of times the output is in O for both S and S' respectively. Using these proportions and the Clopper-Pearson confidence intevals (Clopper & Pearson, 1934), we calculate upper and lower bounds on the probabilities $\mathbb{P}\left(\mathcal{A}(S) \in O\right)$ (denoted as p_0^ℓ, p_0^u) and $\mathbb{P}\left(\mathcal{A}(S') \in O\right)$ (denoted as p_1^ℓ, p_1^u) and subsequently estimate the lower bound ε^ℓ using the following equation:

$$\varepsilon^\ell = \max\left\{ \frac{p_1^\ell - \delta}{p_0^u}, \frac{p_0^\ell - \delta}{p_1^u} \right\}. \tag{1}$$

Algorithm 1 Private prediction framework

```
Input: Training dataset $S = (X, Y)$, teacher count $k$, query dataset $Q$, per-query noise scale $\sigma$
Params: Number of partitions $m$
Initialize PREDICTIONS = []
Training Phase:
$P$ = PARTITION($S, m$)
for $i = 1$ to $m$ do
    $T_i$ = TRAINTEACHER($P_i$)
end for
Prediction Phase:
for $q \in Q$ do
    $T$ = CHOOSE($T_1^m, k, q$)
    $\mathrm{H}_S$ = GETPREDICTIONS($T, q$)
    $\widetilde{\mathrm{H}}_S$ = ADDNOISE($\mathrm{H}_S, \sigma$)
    PREDICTIONS.append(argmax $\widetilde{\mathrm{H}}_S$)
end for
Output: PREDICTIONS
```

The choice of the neighbouring datasets and the output set partly hinges on the constraints imposed on an adversary, influencing the strength of the calculated lower bound ε^ℓ.

Attack interpretation. We can also view auditing as a hypothesis test, where the adversary resolves:

H_0 : S was used to generate the output of the algorithm $\mathcal{A}$

H_1 : S' was used to generate the output of the algorithm $\mathcal{A}$,

using the output of the mechanism. For a rejection region O, let the false positive ($\mathbb{P}(\mathcal{A}(S) \in O)$) and false negative ($\mathbb{P}(\mathcal{A}(S') \in O')$) rates for this hypothesis test be FP and FN respectively. Then, Eq. (1) can be reformulated using upper bounds on FP and FN as:

$$\varepsilon^\ell = \max\left\{ \frac{1 - \delta - FN^u}{FP^u}, \frac{1 - \delta - FP^u}{FN^u} \right\}.$$

3. Private Prediction

In this section, we define privacy-preserving prediction, integrate known algorithms into a unified framework, and present two prevalent methods for reporting the privacy guarantee in this framework. Following the setup of (Dwork & Feldman, 2018), let $\mathcal{A}(S)$ denote a prediction interface produced by applying algorithm $\mathcal{A}$ to the training set S, which produces an output in $\mathcal{Y}$ when queried with a point in $\mathcal{X}$. Let $(\mathcal{A}(S) \leftrightarrow Q)$ denote the sequence of outputs of the interface when queried with a test query sequence Q.

Definition 3 (Private Prediction Interface)**.** A prediction interface $\mathcal{A}(S)$ satisfies $(\alpha, \varepsilon_\alpha)$-RDP if for every test query sequence Q, the output $(\mathcal{A}(S) \leftrightarrow Q)$ satisfies $(\alpha, \varepsilon_\alpha)$-RDP with respect to dataset S.

Alg. 1 gives a general framework for private prediction, based on the Sample Aggregate framework (Nissim et al., 2007). It first divides the training dataset into m splits, training a distinct teacher on each split. In the prediction phase, for each query, it selects k teachers from the m available, aggregating their predictions on the query point q into a histogram H_S. After adding noise to create $\widetilde{\mathrm{H}}_S$, the label that achieves plurality, i.e., the label with the highest count in $\widetilde{\mathrm{H}}_S$, becomes the prediction for q. The added noise typically follows a Gaussian distribution with scale σ, chosen to satisfy the desired privacy guarantee. Mathematically,

$$\textsc{Predictions} = \arg\max_{y \in \mathcal{Y}} \{\mathrm{H}_S + \mathsf{N}(0, \sigma^2)\}. \quad (2)$$

The privacy guarantee of Alg. 1 hinges on the noisy argmax (Eq. (2)) mechanism's privacy properties. We analyze the privacy for each query individually using RDP and combine them using composition theorems. Working with RDP offers tighter composition properties than approximate DP and it facilitates data-dependent privacy leakage analysis (see below). The RDP guarantee is calculated in two ways:

Data Independent. This approach overlooks the histogram's post-processing into a single label release and assumes the release of the entire noisy histogram. It results in conservative privacy accounting, assuming a worst-case scenario for each query and disregards any privacy amplification due to post-processing.

Data Dependent. Proposed in Papernot et al. (2018, Appendix A), this accounting technique applies data-dependent analysis whereby the prediction interface incurs a smaller privacy cost when many teachers agree on a label. It partly accounts for post-processing for these easy queries, which improves accounting. In this case, the privacy parameter itself may release some private information. However, we can use the smooth sensitivity mechanism (Papernot et al., 2018, Appendix B) to release it privately.

4. Auditing Private Prediction

We now present our framework for auditing private prediction. Like Nasr et al. (2021), we follow an *adversary instantiation* approach; we define different adversaries based on their training data altering capabilities and the freedom to choose the test queries. Then, we discuss how we use the results of an adversary's attack as input to audit the RDP guarantees provided by the noisy argmax, which is the core privacy primitive in private prediction. Fig. 1 shows our framework, which we divide into three parts.

1. *Crafter.* Starting with a dataset S, the crafter generates a new dataset S', differing from S by a single data point. These datasets then become the training inputs for private prediction algorithms. We recognize two types of crafters based on their methods of constructing S' from S:
    - Natural Crafter: Adds an in-distribution point.
    - Poisoning Crafter: Adds an adversarial point.
2. *Private Prediction Algorithm.* The private prediction framework $\mathcal{A}(\cdot)$, detailed in Alg. 1, accepts datasets S and S' to create private prediction interfaces $\mathcal{A}(S)$ and $\mathcal{A}(S')$, by passing the datasets through the *Training Phase*. These interfaces then respond to a sequence of queries $Q = (q_1, q_2, \ldots,)$ with sequences $(\mathcal{A}(S) \leftrightarrow Q) = (\mathcal{A}(S; q_1), \mathcal{A}(S; q_2), \ldots,)$ and $(\mathcal{A}(S') \leftrightarrow Q) = (\mathcal{A}(S'; q_1), \mathcal{A}(S'; q_2), \ldots,)$. We audit PATE, CaPC, PromptPATE and Private-kNN.
3. *Distinguisher.* The distinguisher chooses the query sequence Q, observes the output of the interfaces $(\mathcal{A}(S) \leftrightarrow Q)$ and $(\mathcal{A}(S') \leftrightarrow Q)$, and estimates the privacy leakage of the prediction framework by comparing the output distributions. We study two distinguishers based on their query selection methods:
    - Natural Distinguisher: Chooses queries (Q) from a natural distribution, mimicking real-world scenarios, simulated using the test dataset.
    - Adversarial Distinguisher: Chooses Q adversarially, which, in all cases we consider, is querying the interface with the same query q repeatedly.

4.1. Adversaries

Using the above framework, we define three adversaries using the aforementioned crafters and the distinguishers.

Nat-AdvQ. Natural crafter, adversarial distinguisher, simulating a membership inference adversary who tries to infer the membership of an in-distribution training set example, but who adversarially chooses queries.

Pois-AdvQ. Poisoning crafter, adversarial distinguisher, simulating a stronger adversary with the power to choose a worst case poisoning point to maximize the efficacy of a membership inference attack.

Pois-NatQ. Poisoning crafter, natural distinguisher, where the adversary statically poisons S to form S', but can only use natural queries to distinguish. The responses of a private prediction interface on such queries facilitate training a student model (as suggested in the original formulation of PATE, CaPC and PromptPATE), which can then be used to answer queries indefinitely without any additional privacy cost. The privacy leakage of the student model is bounded by the privacy leakage of **Pois-NatQ** due to data processing inequality, though student training may not reduce leakage (Jagielski et al., 2023). The restriction to natural queries is important since with a reasonably small privacy budget, a performant student model can only be trained on a natural set of queries.

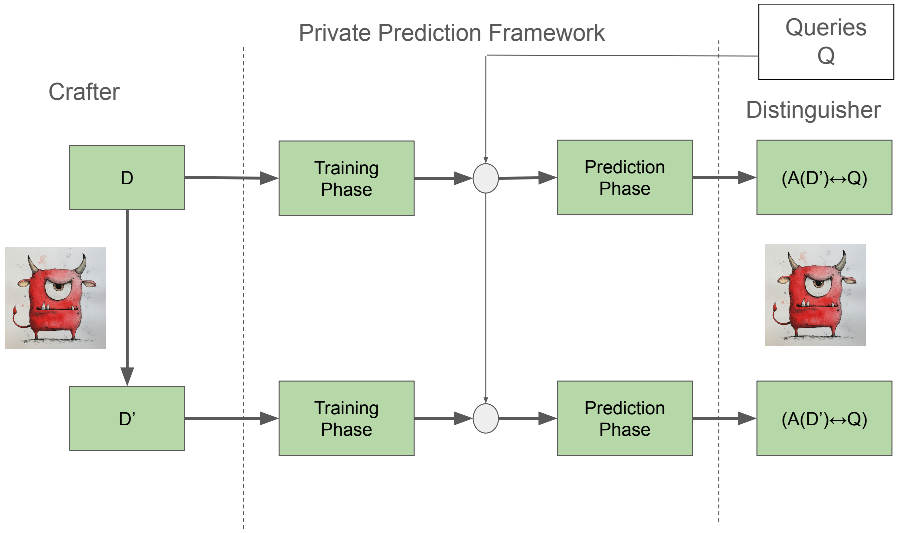


Figure 1. Framework to audit private prediction algorithms.

4.2. Auditing RDP of Noisy Argmax

The standard approach to auditing would take the full output of the private prediction algorithm on the entire set of queries, and perform an attack using this sequence of outputs. However, the space of outputs is very high dimensional: with n_Q queries, and C total classes, the total output space is an enormous C^{n_Q}. To condense this output space, we instead audit each query in isolation, and compose the lower bounds we obtain for each query. However, composition theorems in (ε, δ)-DP are lossy and composing (ε, δ)-DP lower bounds is invalid.[2] Therefore, we must audit the per-query outputs in Renyi DP to be able to use its lossless composition properties.

We now present an approach to audit RDP and an upper bound method to calculate the exact Renyi divergence between neighboring histograms.

Auditing with the 2-cut. Renyi Divergence lacks a hypothesis testing interpretation (Balle et al., 2020), meaning that in general, there may not exist membership inference attacks, that can tightly audit an RDP guarantee. However, Balle et al. (2020) show that the 2-cut of the Renyi divergence, which calculates the supremum of the Renyi divergence between induced bernoulli distributions over all possible sets of the output space has a hypothesis testing interpretation, and is a lower bound on the Renyi divergence between the distributions. The 2-cut of the Renyi divergence between two distributions μ_1 and μ_2 is defined as

$$\overline{D_\alpha}^{2}(\mu_1||\mu_2) :=$$

$$\sup_{O \subseteq \mathcal{O}} \frac{1}{\alpha - 1} \log\left(p_1^\alpha p_2^{1-\alpha} + (1-p_1)^\alpha (1-p_2)^{1-\alpha}\right), \quad (3)$$

where $p_1 = \mathbb{P}(\mu_1 \in O)$ and $p_2 = \mathbb{P}(\mu_2 \in O)$, and we have $\overline{D_\alpha}^{2}(\mu_1||\mu_2) \leq D_\alpha(\mu_1||\mu_2)$. Thus, we can lower bound the RDP guarantee of a mechanism by lower bounding the 2-cut of the Renyi divergence between output distributions generated by neighboring datasets using the FP and FN rates of any membership inference attack by choosing O to be the rejection region. To ensure statistical validity, we use Clopper Pearson confidence intervals on the results of a Monte Carlo simulation to bound each term in Eq. (3), with 95% confidence. This lower bound is statistically valid for all sample sizes owing to the validity properties of Clopper Pearson intervals.

We propose three more RDP auditing approaches in Appendix B, but focus on the 2-cut audit here since it is always valid and has an attack interpretation.

An Improved Upper Bound - Exact Renyi Divergence We observe that, given a fixed histogram $H = [n_1, n_2, \ldots, n_C]$, we can compute the probability that the noisy argmax returns a given class c as:

$$\Pr[c] = \int_{-\infty}^{\infty} \phi\left(\frac{x - n_c}{\sigma}\right) \prod_{i \neq c} \Phi\left(\frac{x - n_i}{\sigma}\right) dx, \quad (4)$$

where ϕ and Φ are the probability density and cumulative distribution functions of the standard normal distribution. From these probabilities, we can directly compute the exact Renyi divergence between neighboring histograms. This direct calculation can be used as an upper bound for a given attack, by measuring the Renyi divergence between the histograms resulting from S and S'. Given the neighboring histograms from an attack, this exact calculation is the tightest possible calculation for the privacy leakage and hence better than previous data-dependent

[2] This issue is avoided in DP-SGD audits, since the output is the model and accounting for composition over queries is not needed.

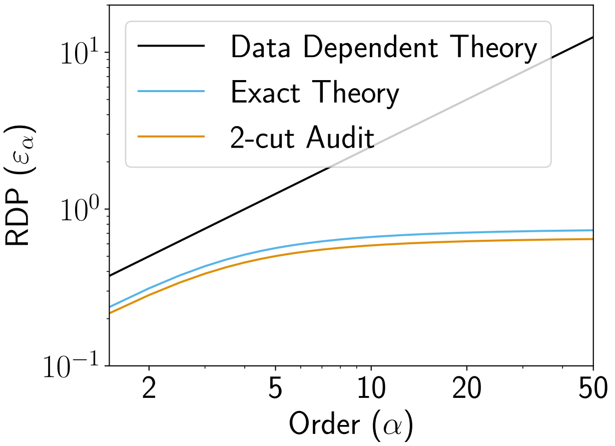


Figure 2. RDP audit for noisy argmax

analysis. In our experiments, we also plot this bound to separate the looseness in analysis and looseness due to adversary capabilities. Using this computation as a primitive, it is possible to get a faithful data-independent bound by taking the supremum of the Renyi divergence over all possible neighboring histograms. However, we do not investigate this, as the space of neighboring histograms is combinatorially large; we only use the bound for given histograms as an upper bound for the best performing audit (see Sec. 7 for future work directions).

Fig. 2 shows the results of a 2-cut audit, exact Renyi divergence, and data-dependent theoretical calculations for a noisy argmax mechanism on a synthetic histogram, plotted against Renyi divergence orders. The exact calculation significantly outperforms the prior theoretical estimation, and the 2-cut audit closely approximates the exact results.

5. Auditing PATE, CaPC and PromptPATE

We now apply our auditing framework to three private prediction algorithms which share a similar design: PATE, CapC and PromptPATE. We describe each algorithm using the framework in Alg. 1, outline the experimental setup, plot the audit results in Fig. 3 and 4 and discuss key findings.

Each private prediction has a low signal-to-noise ratio due to the low privacy cost per query. For this reason, we need roughly 10^8 experiments per query to produce a reasonable privacy lower bound. If each of these experiments required k full model trainings, our auditing would be computationally impractical. Therefore, we introduce parametric modeling assumptions to mimic the randomness in the *training phase*, providing an estimated distribution of the H_S (and $\mathrm{H}_{S'}$) histogram across independent experiments, which we can directly sample from (this strategy is also used in Wang et al. (2022) to facilitate test-time attacks on PATE).

For all experiments, we use 200 or 250 teachers and Gaussian noise with σ in $\{20, 25, 30, 40\}$, based on values used in prior work and to ensure a reasonable accuracy and privacy cost. We defer exact hyperparameter details to Appendix D.

5.1. PATE (Papernot et al., 2016; 2018)

In the training phase of PATE, we partition the input dataset S into k subsets randomly and train a teacher model on each. During the prediction phase, we aggregate each trained teacher's predictions for a query q in a histogram, and output a prediction using the noisy argmax mechanism.

Parametric Assumption. For a query q, let P_q denote the distribution over classes which generates the predictions of the k teachers trained on equally sized random subsets of the training set S and let P'_q denote the distribution over classes which generates the predictions of a teacher trained on a random subset of the training set S with the datapoint (x', y') always included. Then, we model the histograms H_S and $\mathrm{H}_{S'}$ as:

$$\mathrm{H}_S(q) = \texttt{Mult}(k, P_q)$$
$$\mathrm{H}_{S'}(q) = \texttt{Mult}(k-1, P_q) + \texttt{Mult}(1, P'_q),$$

where $\texttt{Mult}(n, \mu)$ denotes a sample of the multinomial distribution with n trials. We estimate P_q and P'_q by running N_g instantiations of the training phase and using the maximum likelihood estimate of the resulting predictions.

Experiment Setup. We audit PATE on the MNIST, CIFAR10 and Fashion MNIST (Xiao et al., 2017) datasets. The **Nat-AdvQ** adversary augments S with a test query to create S', then repeatedly queries the interface with it. Both **Pois-AdvQ** and **Pois-NatQ** adversaries add a mislabelled test query (labeled with the second most commonly produced label) to S forming S'; the former queries the interface repeatedly with the same point, while the latter uses natural queries, which we model using the test set. Using gradient matching techniques like those in Geiping et al. (2020), we tried finding better poisoning points which can impact teacher predictions for many queries for **Pois-NatQ**, but they didn't outperform simple mislabelled points. Designing stronger poisoning attacks is an interesting opportunity for future work to improve our audits.

5.2. CaPC (Choquette-Choo et al., 2020)

The CaPC framework closely resembles PATE, with a key distinction: data division into k teachers occurs deterministically, not randomly, as it is designed for a multiparty setting where each party has a fixed local dataset. The remainder of CaPC is identical — we train teacher models on each data subset, and aggregate teacher predictions with a noisy argmax to get the private prediction.

Parametric Assumption. For a query q, let P_q^i denote the distribution over classes which generates the predictions of the teacher i trained on $S_i \subset S$ and let $P_q^{1'}$ denote the distribution over classes which generates the predictions of a teacher trained on $S_1 \cup (x', y')$, where the first teacher is chosen without loss of generality. Then, we model the histogram H_S and H'_S as:

$$\mathrm{H}_S(q) = \sum_{i=1}^{k} \texttt{Mult}(1, P_q^i)$$
$$\mathrm{H}_{S'}(q) = \texttt{Mult}(1, P_q^{1'})) + \sum_{i=2}^{k} \texttt{Mult}(1, P_q^i).$$

We estimate P_q^i and $P_q^{1'}$ by running N_g instantiations of the training phase and using the maximum likelihood estimate of the resulting predictions.

Experiment Setup. We do not change this from PATE.

5.3. PromptPATE (Duan et al., 2023)

PromptPATE is an In Context Learning based variant of PATE. In this case, the training dataset is divided into k subsets and the examples in each subset are used in the few-shot prompt to a pretrained language model. For each teacher, the prompt consists of the task description and an example. In the prediction phase, we prompt the language model with the incoming query appended to the corresponding teacher prompt and generate the prediction for that query. The final prediction is output by applying noisy argmax to the histogram of teacher predictions.

Parametric Assumption. In this case, all the steps of the training phase are deterministic since we sample from an LLM with temperature = 0 and the only randomness in the mechanism is due to the Gaussian noise added to the histogram. Hence, we don't need to make a parametric assumption to audit PromptPATE.

Experiment Setup. For PromptPATE, we work with the SST2 (Sentiment Classification) (Socher et al., 2013), AGNEWS (Article Classification) (Zhang et al., 2015b), DBPedia (Topic Classification) (Zhang et al., 2015a) and TREC (Question Classification) (Li & Roth, 2002) datasets. We use MISTRAL-7B (Jiang et al., 2023) as the base model and a one-shot prompt it with the task description and an example for each teacher. The **Nat-AdvQ** swaps a train set example with a test query which is used in one of the teacher prompts to produce predictions, and it queries the interface repeatedly with the same query. The **Pois-AdvQ** mirrors this with a mislabeled test query as the poison. For **Pois-NatQ**, we test four different adversarial prompts, for instance, forcing a particular prediction, and report the leakage due to the best adversary for each dataset in the results. We describe the different prompts and their performance on each dataset in detail in Appendices D and E.

5.4. Experimental Results

Fig. 3 and 4 plot the results of our audit for the three algorithms and the three adversaries we consider along with the data dependent theoretical upper bound and the exact RDP calculation. While we perform the privacy analysis and the audits in RDP, we plot the privacy leakage in terms of ε for $\delta = 10^{-6}$ using Theorem 1. While this conversion is not valid for audits since it is an upper bound on the privacy loss, we choose to plot the converted ε values for ease of illustration and defer the raw RDP audit plots to Appendix E. For all algorithms, we instantiate both **Nat-AdvQ** and **Pois-AdvQ** for n_Q different queries and plot the result of the audit for the queries leaking most privacy as it is a proxy for the worst case privacy leakage in each scenario.

Privacy analysis is not tight. The gap between the exact RDP calculation (Sec. 4.2) and the data dependent theoretical calculation (from Papernot et al. (2018)) in all plots in Fig. 3 and 4 highlights room for improvement in the data dependent privacy analysis. Moreover, the lower privacy cost for natural queries compared to adversarial queries points to potential improvements in analysis under distributional assumptions on queries to a private prediction interface.

Impact of poisoning capability on privacy leakage. We compare the privacy leakage due to poisoning across two axes: algorithms and adversaries. Fig. 3 shows that relative to exact RDP calculations, both **Nat-AdvQ** and **Pois-AdvQ** show the least privacy leakage in PATE, followed by CaPC and PromptPATE. Both adversaries exhibit similar leakage levels for CaPC and PromptPATE. However, **Pois-AdvQ** compromises privacy more than **Nat-AdvQ** in PATE. This is in line with the ease of poisoning in different scenarios. For PromptPATE, each teacher contains a single data point, making it easy to deterministically change a teacher's vote by adding a correctly labelled query to the prompt of a teacher which initially misclassified the query (**Nat-AdvQ**), or by adding a mislabelled query to the prompt of a teacher which correctly classified the query (**Pois-AdvQ**). This leads to tight audits and nearly maximal privacy leakage. Likewise, in CaPC, the deterministic selection of data subsets for each teacher allows for consistent vote flip to a query on neighboring datasets by adding a specific example. However, PATE introduces additional randomness through data shuffling, diminishing the predictability of flipping a teacher's vote by a data point change. This effect is more pronounced in in-distribution membership inference (**Nat-AdvQ**) than in adversarial membership inference (**Pois-AdvQ**), as the inclusion of an adversarial

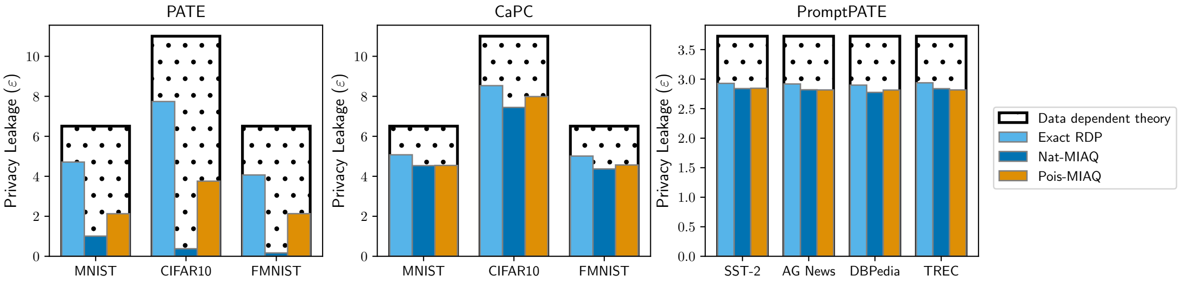


Figure 3. Privacy leakage of PATE, CaPC and PromptPATE for adversaries with adversarial query capability.

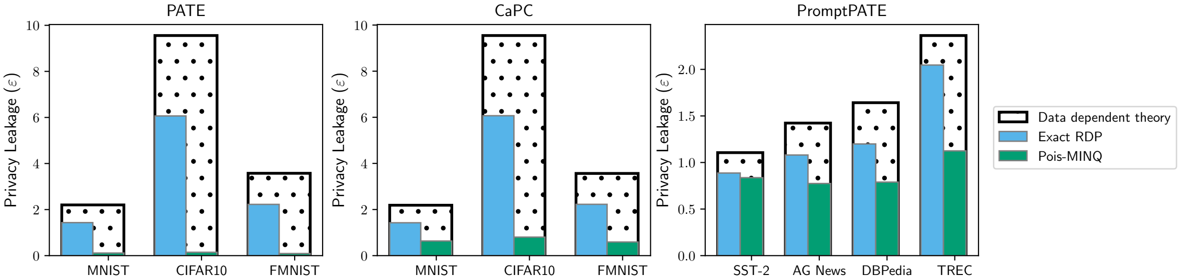


Figure 4. Privacy leakage of PATE, CaPC and PromptPATE for adversaries restricted to making natural queries.

(mislabelled) point is more likely to change the vote of any teacher than a natural point.

Impact of query capability on privacy leakage. A comparison of the privacy leakage in Fig. 3 and 4 shows that privacy leakage due to **Pois-NatQ** is much lower than the privacy leakage due to **Nat-AdvQ** and **Pois-AdvQ** across all algorithms. This stems from two major factors: 1. natural queries inherently incur lower privacy costs due to the presence of "easy" queries where teachers concur, as the lower theoretical and exact values in Fig. 4 compared to Fig. 3 corroborate, and 2. poisoning teachers to change their responses on multiple queries is more challenging, evidenced by the gaps between the audit and exact RDP calculations in Fig. 4. Moreover, the privacy leakage of **Pois-NatQ** across algorithms also follows the order PATE $<<$ CaPC $<$ PromptPATE, owing to the relative poisoning difficulties. In fact for PATE, there is negligible privacy leakage, whereas for PromptPATE, the ease of crafting prompts to permanently change a teacher's behavior on all queries leads to significant privacy leakage. Lastly, we note that the adversarially crafted queries in the cases we consider are repeated queries. We can limit the privacy leakage in such cases using cached outputs for repeated queries of the same point.

6. Auditing Private-kNN (Zhu et al., 2020)

For Private-kNN, we treat each datapoint as a teacher and take the prediction of each datapoint as its label. In the prediction phase, we subsample the teachers (datapoints) with probability γ each and return the top k closest points to the query as the chosen set of teachers. Then, we aggregate the teacher votes (labels) into a histogram and apply the noisy argmax mechanism to produce a prediction.

Parametric Assumption. Let P_q denote the distribution of the top-k nearest neighbours and let P_q^k denote the distribution of the last (k-th) nearest neighbour, of a query q when the training points are subsampled independently with probability γ. Also, let (x', y') be the poisoned point. Then, we model the histograms as:

$$\mathrm{H}_S(q) = \texttt{Mult}(k, P_q)$$
$$\mathrm{H}_{S'}(q) = \texttt{Mult}(k, P_q) + \texttt{Ber}(\nu)(\mathbf{1}\{y'\} - \texttt{Mult}(1, P_q^k))),$$

where ν denotes the probability that (x', y') is included in the top-k nearest neighbours. Let x' be the r_q-th closest point to the query q amongst the training set. Then,

$$\nu = \begin{cases} \gamma & \text{if } r_q \leq k, \\ \sum_{i=0}^{k-1} \binom{r_q - 1}{i} \gamma^{i+1} (1-\gamma)^{r_q - i - 1} & \text{if } r_q > k. \end{cases}$$

Experiment setup For Private-kNN, we work with the CIFAR10, Fashion MNIST, SST2 and AGNEWS datasets. We use the ViT-L/16 models pretrained on ImageNet21k to extract the features for image classification datasets and the RoBERTa-Large model to extract the features for the text classification datasets. As with other algorithms, the

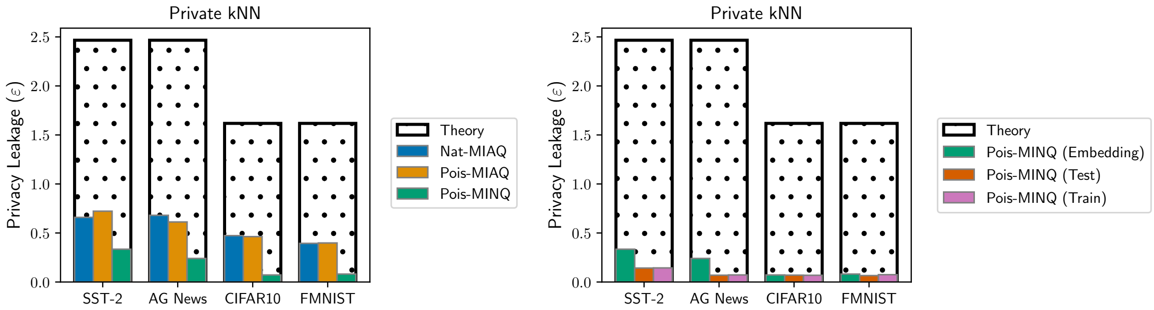


Figure 5. Privacy leakage for Private-kNN

Nat-AdvQ (**Pois-AdvQ**) adds a (mislabelled) test point and repeatedly queries the same point. For **Pois-NatQ**, we consider three different adversaries which find a point maximizing the probability of being in the top-k for a given set of natural queries, from the test set, from the train set and from the whole feature space, respectively.

Fig. 5 shows the results of auditing Private-kNN. Since the theory calculation here is data-independent, there is a big gap between the privacy leakage even for the strongest adversaries and the theoretical bound. Echoing the trends of PATE-like algorithms, the privacy leakage for **Pois-AdvQ** and **Nat-AdvQ** is similar and it is higher than the leakage for **Pois-NatQ**, likely due to similar reasons. The right plot of Fig. 5 highlights the difference in privacy leakage for an adversary constrained to mislabel a point from the datasets (test or train) against an adversary who is allowed to choose an adversarial embedding to poison the dataset. The additional access to the embedding space, as opposed to natural input space, results in a doubling of privacy leakage in text classification datasets.

7. Future Work

One of the main contributions of our work is identifying several opportunities to improve the analysis of private prediction algorithms (or empirical reasons for looseness in audits). These range from entirely empirical observations to concrete mathematical questions.

Improved Theoretical Analysis. Our exact Renyi divergence computation avoids a pessimistic union bound in Prop. 8 of (Papernot et al., 2018). However, evaluating this integral for all possible neighbouring histograms appears computationally intractible with our current approach. A deeper characterization of the integral Eq. (4) may remove or significantly lighten the computational burden. Furthermore, private k-NN uses amplification by subsampling for its privacy analysis. The looseness of our audits point towards potential improvements using data-dependent analysis techniques which incorporate subsampling.

Stronger Attacks. Natural Queries uniformly reduce measured privacy leakage. This points towards a potential for tighter privacy analysis under distributional assumptions on the queries. However, this could also be the result of weak attacks—future work might design stronger poisoning attacks capable of attacking multiple natural queries with a small number of poisoning examples. Incorporating group privacy in an audit may lead to tighter bounds due to the potential of designing stronger adversaries who can affect the output on multiple natural queries by changing multiple training datapoints. See Appendix C for an outline of group privacy based auditing.

Amplification due to random partitioning. PATE involves randomly dividing the data into partitions, which appear to empirically improve privacy leakage. It may be possible to also take advantage of this source of randomness. Otherwise, stronger poisoning attacks to increase measured leakage would need to be robust to this randomness.

Limitations. Our auditing methods may be improved by improved statistical techniques, which can in turn improve the tightness of audits or reduce the computational burden for achieving similar tightness. Better data poisoning techniques capable of attacking multiple queries can improve tightness of natural query audits.

Impact Statement

Our work may be used to justify privacy parameters which prevent our attacks, but which are vulnerable to stronger, currently unknown attacks. However, measuring effectiveness of our attacks is an important step to improving both analysis and attacks.

Acknowledgements

Karan Chadha was partially supported by ONR N00014-22-1-2669, DAWN Consortium and NSF: NSF 2006777. We thank Thomas Steinke, Borja Balle, and Andreas Terzis for comments on our work.

Contributions

Matthew Jagielski and Nicolas Papernot ideated the project. Karan Chadha, Matthew Jagielski and Nicolas Papernot refined the project idea and scope, and designed the experimental plan. Karan Chadha wrote the code for the experiments. Matthew Jagielski helped with running experiments on the infrastructure. Christopher A. Choquette-Choo and Milad Nasr provided feedback and advice on refining the experimental plan. Everyone wrote the paper.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Andrew, G., Kairouz, P., Oh, S., Oprea, A., McMahan, H. B., and Suriyakumar, V. One-shot empirical privacy estimation for federated learning. *arXiv preprint arXiv:2302.03098*, 2023.

Andrews, D. W. Inconsistency of the bootstrap when a parameter is on the boundary of the parameter space. *Econometrica*, pp. 399–405, 2000.

Ba, A. D., Lo, G. S., and Ba, D. Divergence measures estimation and its asymptotic normality theory using wavelets empirical processes i. *Journal of Statistical Theory and Applications*, 17(1):158–171, 2018.

Balle, B., Barthe, G., and Gaboardi, M. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31, 2018.

Balle, B., Barthe, G., Gaboardi, M., Hsu, J., and Sato, T. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 2496–2506. PMLR, 2020.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pp. 464–473. IEEE, 2014.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Choquette-Choo, C. A., Dullerud, N., Dziedzic, A., Zhang, Y., Jha, S., Papernot, N., and Wang, X. Capc learning: Confidential and private collaborative learning. In *International Conference on Learning Representations*, 2020.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In *International conference on machine learning*, pp. 1964–1974. PMLR, 2021.

Choquette-Choo, C. A., McMahan, H. B., Rush, K., and Thakurta, A. Multi-epoch matrix factorization mechanisms for private machine learning. *arXiv preprint arXiv:2211.06530*, 2022.

Choquette-Choo, C. A., Ganesh, A., McKenna, R., McMahan, H. B., Rush, K., Thakurta, A. G., and Xu, Z. (amplified) banded matrix factorization: A unified approach to private training. *arXiv preprint arXiv:2306.08153*, 2023a.

Choquette-Choo, C. A., Ganesh, A., Steinke, T., and Thakurta, A. Privacy amplification for matrix mechanisms. *arXiv preprint arXiv:2310.15526*, 2023b.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Denisov, S., McMahan, H. B., Rush, J., Smith, A., and Guha Thakurta, A. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *Advances in Neural Information Processing Systems*, 35: 5910–5924, 2022.

Ding, Z., Wang, Y., Wang, G., Zhang, D., and Kifer, D. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 475–489, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Duan, H., Dziedzic, A., Papernot, N., and Boenisch, F. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *arXiv preprint arXiv:2305.15594*, 2023.

Dwork, C. and Feldman, V. Privacy-preserving prediction. In *Conference On Learning Theory*, pp. 1693–1702. PMLR, 2018.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Efron, B. and Tibshirani, R. J. *An introduction to the bootstrap*. CRC press, 1994.

Geiping, J., Fowl, L. H., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2020.

Goodman, L. A. On simultaneous confidence intervals for multinomial proportions. *Technometrics*, 7(2):247–254, 1965.

Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020.

Jagielski, M., Nasr, M., Choquette-Choo, C., Lee, K., and Carlini, N. Students parrot their teachers: Membership inference on model distillation. *arXiv preprint arXiv:2303.03446*, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021.

Katz, D., Baptista, J., Azen, S., and Pike, M. Obtaining confidence intervals for the risk ratio in cohort studies. *Biometrics*, pp. 469–474, 1978.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kong, W., Medina, A. M., and Ribero, M. R\'enyitester: A variational approach to testing differential privacy. *arXiv preprint arXiv:2307.05608*, 2023.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, X. and Roth, D. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://www.aclweb.org/anthology/C02-1150.

Li, Z. and Zhang, Y. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 880–895, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lu, F., Munoz, J., Fuchs, M., LeBlond, T., Zaresky-Williams, E., Raff, E., Ferraro, F., and Testa, B. A general framework for auditing differentially private machine learning. *Advances in Neural Information Processing Systems*, 35:4165–4176, 2022.

Malek Esmaeili, M., Mironov, I., Prasad, K., Shilov, I., and Tramer, F. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34:6934–6945, 2021.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Nasr, M., Songi, S., Thakurta, A., Papernot, N., and Carlin, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pp. 866–882. IEEE, 2021.

Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., and Terzis, A. Tight auditing of differentially private machine learning. *arXiv preprint arXiv:2302.07956*, 2023.

Nissim, K., Raskhodnikova, S., and Smith, A. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 75–84, 2007.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, U. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9312–9321, 2021.

Pillutla, K., Andrew, G., Kairouz, P., McMahan, H. B., Oprea, A., and Oh, S. Unleashing the power of randomization in auditing differentially private ml. *Advances in Neural Information Processing Systems*, 36, 2024.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models (s&p'17). 2016.

Sison, C. P. and Glaz, J. Simultaneous confidence intervals and sample size determination for multinomial proportions. *Journal of the American Statistical Association*, 90(429):366–369, 1995.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Steinke, T., Nasr, M., and Jagielski, M. Privacy auditing with one (1) training run. *arXiv preprint arXiv:2305.08846*, 2023.

Tramèr, F., Shokri, R., San Joaquin, A., Le, H., Jagielski, M., Hong, S., and Carlini, N. Truth serum: Poisoning machine learning models to reveal their secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2779–2792, 2022.

Tramer, F., Terzis, A., Steinke, T., Song, S., Jagielski, M., and Carlini, N. Debugging differential privacy: A case study for privacy auditing. *arXiv preprint arXiv:2202.12219*, 2022.

Wang, J., Schuster, R., Shumailov, I., Lie, D., and Papernot, N. In differential privacy, there is truth: on vote-histogram leakage in ensemble private learning. *Advances in Neural Information Processing Systems*, 35:29026–29037, 2022.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., and Zhang, Y. Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*, 2023.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015a. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *NIPS*, 2015b.

Zhu, Y., Yu, X., Chandraker, M., and Wang, Y.-X. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11854–11862, 2020.

Notation	Meaning
ε	Pure DP parameter
(ε, δ)	Approximate DP parameter
ε_α	Renyi DP parameter
β^ℓ	Lower bound on the parameter β
S	Training set
$\mathcal{S}$	Training set space
Q	Query set for private prediction
n_Q	Number of queries
C	Number of classes
k	Number of teachers
N_g	Number of examples to learn generative models
T	Number of experiments for auditing
Nat-AdvQ	Natural membership inference, adversarial queries
Pois-AdvQ	Poisoned membership inference, adversarial queries
Pois-NatQ	Poisoned membership inference, natural queries
H_S	Histogram of aggregated predictions
$\texttt{Mult}(k, p)$	Multinomial Distribution sampling k examples with distribution p
$\texttt{Ber}(p)$	Bernoulli Distribution with parameter p

Table 1. Notation

A. Notation

B. Auditing RDP guarantees of noisy argmax

In this section, we propose additional methods for auditing the RDP guarantee of the noisy argmax mechanism. The first of these methods uses the k-cut of the Renyi Divergence, which is a generalization of the 2-cut based method proposed in Sec. 4.2. The remaining two methods are based on using bootstrap to estimate the 2-cut (or the k-cut) directly instead of relying on estimates of probability distributions which are then used to give lower bounds on the RDP. We end the section by discussing the auditing performance, assumptions needed for validity and accuracy for these lower bounds.

First, for completeness, we spell out the equation we use to lower bound the 2-cut. Let p_1, p_2 denote $\mathbb{P}(\mu_1 \in O)$ and $\mathbb{P}(\mu_2 \in O)$ respectively and let p_j^ℓ, p_j^u denote the Clopper Pearson lower and upper bounds on p_j. Then, we lower bound the 2-cut as:

$$\overline{D_\alpha}^2(\mu_1||\mu_2) \geq \frac{1}{\alpha - 1} \log \left((p_1^\ell)^\alpha (p_2^u)^{1-\alpha} + (1 - p_1^u)^\alpha (1 - p_2^\ell)^{1-\alpha} \right).$$

Auditing with the k-cut. Even the best 2-cut lower bounds may not be tight as they lose information when restricting to a particular output set. Balle et al. (2020) also introduced the notion of k-cut of the Renyi divergence, a generalization of the 2-cut, which calculates the supremum of the Renyi divergence between induced distributions over all possible k-sized partitions of the output set and is a lower bound on the Renyi divergence between the distributions. The k-cut of the Renyi divergence between two distributions μ_1 and μ_2 is defined as

$$\overline{D_\alpha}^k(\mu_1||\mu_2) := \sup_{\substack{O_1, \ldots, O_k \subseteq \mathcal{O} \\ O_i \cap O_j = \phi \\ \cup_{i=1}^n O_i = \mathcal{O}}} \frac{1}{\alpha - 1} \log \left(\sum_{i=1}^k p_1(i)^\alpha p_2(i)^{1-\alpha} \right), \tag{5}$$

where $p_1(i) = \mathbb{P}(\mu_1 \in O_i)$ and $p_2(i) = \mathbb{P}(\mu_2 \in O_i)$. with $\overline{D_\alpha}^k(\mu_1||\mu_2) \leq D_\alpha(\mu_1||\mu_2)$. When the output space itself is discrete with cardinality k, the k-cut is equivalent to the Renyi divergence. Thus, we can lower bound the RDP guarantee of a mechanism by lower bounding the k-cut of the Renyi divergence between output distributions generated by neighboring datasets. To ensure statistical validity, we use simultaneous confidence interval techniques (Goodman, 1965; Sison & Glaz, 1995) on the results of a Monte Carlo simulation to get asymptotically valid lower bounds. Note that compared to the 2-cut based method. Let $p_j^\ell(i), p_j^u(i)$ denote the lower and upper bounds on $p_j(i)$. Using these, we lower bound the k-cut as:

$$\overline{D_\alpha}^k(\mu_1||\mu_2) \geq \frac{1}{\alpha-1}\log\left(\sum_{i=1}^{k}(p_1^\ell(i))^\alpha(p_2^u(i))^{1-\alpha}\right).$$

Note that since the k-cut of a Renyi divergence doesn't have a hypothesis testing interpretation, the k-cut audit doesn't admit an attack interpretation, i.e. for a mechanism, there may not be any membership attack with RDP leakage equal to that calculated by a k-cut audit. While the k-cut captures more information than the 2-cut due to increased granularity, the weaker validity properties of (both Goodman and Sison-Glasz) simultaneous confidence intervals compared to Clopper Pearson intervals make it hard to determine a-priori which of the two techniques would result in a tighter audit. We focus on the 2-cut audit for the main paper, the audits have similar performance.

Bootstrap based methods. In both the 2-cut and the k-cut audit, we estimate the output proportions within certain sets (or multiple sets for k-cut) and subsequently apply a non-linear transformation using the Renyi divergence to align with theoretical calculations. However, this transformation might introduce looseness. We can potentially obtain tighter lower bounds on the true Renyi divergence by directly deriving confident bounds for the 2-cut or the k-cut. Lu et al. (2022) demonstrate this approach for pure differential privacy ($\delta = 0$), employing Log-Katz (Katz et al., 1978) intervals to derive lower bounds on the log ratio of proportions, bypassing the individual proportion bounds usually obtained through Clopper Pearson and the subsequent log ratio computation. Given the absence of superior confidence intervals for the Renyi divergence functional applied to proportions, we adopt the bootstrap method (Efron & Tibshirani, 1994) for lower bound estimation. This method is suitable for any functional that satisfies asymptotic normality, a property satisfied for the Renyi divergence measure as shown in Ba et al. (2018).

- **2-cut Bootstrap audit**: We estimate the 2-cut of the Renyi divergence between two distributions using Monte Carlo simulations for a chosen output set. Then, we use bootstrap resampling to estimate the distribution of the 2-cut of the Renyi divergence between the two distributions and use the quantiles of the bootstrap distribution to find an asymptotically valid lower bound.
- **k-cut Bootstrap audit**: We use Monte Carlo simulations to get samples from a categorical distribution over classes and estimate the probability of the output being in each class. Then, we use bootstrap resampling to estimate the distribution of the k-cut of the Renyi divergence between the two distributions and use the quantiles of the bootstrap distribution to find an asymptotically valid lower bound.

A few things are important to note for the bootstrap based audit:

1. Bootstrap methods fail when the true Renyi divergence is 0 since it is on the boundary of the values a Renyi divergence can take.Due to the intrinsic variability of Monte Carlo simulations, it's rare for a pair of samples from an identical distribution to match perfectly. This typically yields a positive $(1-\alpha)$ percentile for the resulting distribution. This phenomenon isn't unexpected, as studies have pointed out the bootstrap's limitations at boundaries, observable even in basic scenarios like gaussian mean estimation (Andrews, 2000).
2. The error rate for bootstrap is generally $O(\frac{1}{\sqrt{n}})$, which implies that when the true Renyi divergence values are smaller than $\frac{1}{\sqrt{n}}$, the results of the bootstrap are unreliable.
3. Even when valid, the lower bounds of bootstrap are strictly valid only for a particular order. However, we observe in practice that using the same samples to generate lower bounds on all orders doesn't cause any issues.

The first two points, in particular, cast doubt on the applicability of bootstrap methods for auditing purposes. Given the minimal Renyi privacy leakage for individual queries in private prediction, the reliability of bootstrap results becomes questionable. Ensuring dependability would necessitate conducting an impractical number of experiments, upwards of 10^{12}.

Fig. 6 shows the performance of all our auditing technique along with the exact theory calculation when applying the noisy argmax mechanism to synthetic neighbouring histograms $[14, 12, 10, 8, 6]$ and $[13, 13, 10, 8, 6]$ with $\sigma = 2$. Based on this figure, both using the k-cut audit over the 2-cut audit and using bootstrap based methods over proportional confidence interval based methods may lead to tighter audits, especially for smaller values of the order α. However, these gains are not consistent across histograms. Due to these reasons and the desirable properties of the 2-cut confidence interval based audit as summarized in Table 2, we use the 2-cut audits for all our main results.

Property Audit	Always Valid	Attack Interpretation	Valid at 0	Valid for all orders
2-cut Confidence Interval	✓	✓	✓	✓
k-cut Confidence Interval	×	×	✓	✓
2-cut Bootstrap	×	✓	×	×
k-cut Bootstrap	×	×	×	×

Table 2. Table of auditing methods and their properties

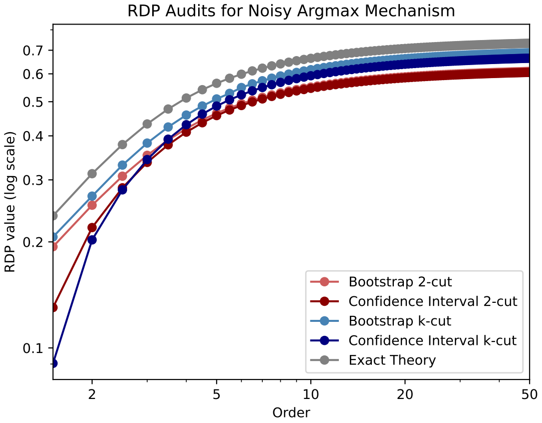


Figure 6. RDP audits for noisy argmax mechanism

C. Group Privacy-based Audit

We first outline a methodology for auditing by changing multiple points using group privacy, highlighting its implications for auditing private predictions. Here, we empower the adversary with the ability to change multiple points but raise the success criteria based on theoretical group privacy guarantees.

Consider a mechanism $\mathcal{A}$ that satisfies $(\alpha, \varepsilon_\alpha)$-DP for some α. This means, for neighboring datasets S and S', the Renyi divergence of order α between the distributions of $\mathcal{A}(S)$ and $\mathcal{A}(S')$ does not exceed ε_α. For datasets S and $S^{(m)}$ differing by m datapoints, the Renyi divergence of order $\alpha/2^m$ between $\mathcal{A}(S)$ and $\mathcal{A}(S')$ is no greater than $3^m\varepsilon_\alpha$. This general estimate may not always be precise. However, specific mechanisms like the Gaussian mechanism offer more exact bounds. For example, adding Gaussian noise with scale σ achieves $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP for all α with an ℓ_2 sensitivity of 1. For datasets differing by m datapoints, the RDP adjusts to $(\alpha, \frac{m^2\alpha}{2\sigma^2})$-RDP.

When studying the privacy leakage due to an adversary which can alter multiple datapoints (say m), we get a valid lower bound on the privacy leakage of the mechanism's group privacy guarantees. These lower bounds can match a group privacy upper bound if all of the following conditions are satisfied:

1. The adversary and/or input dataset is chosen to maximize privacy leakage.
2. The privacy analysis for the mechanism on adjacent datasets is lossless.
3. The group privacy conversion applied to the privacy analysis on adjacent datasets is lossless.

Compared to the standard auditing analysis, group privacy adds an additional condition (3.) to ensure tightness of the lower bounds. Thus, the privacy leakage of the adversary which can alter multiple data points can be converted back to a lower bound on the privacy guarantees of the mechanism if we know the exact functional which characterizes the worst case group privacy loss for mechanism given a privacy guarantee for the mechanism on adjacent datasets.

For private predictions with adversaries unable to control queries, granting them the power to modify multiple datapoints opens a new avenue for studying privacy leakage. However, to determine if such adversaries cause greater leakage than those altering a single datapoint—after normalizing for group privacy—we must identify a precise group privacy calculation for the noisy argmax mechanism. We leave this as an interesting direction of future work and highlight some of our intuitions for particular cases:

- PATE: Designing adversaries that change m examples within a teacher, affecting its vote on at least m^2 natural queries, could yield stronger bounds. This is because group privacy under the Gaussian mechanism worsens as m^2, while privacy degrades linearly with m during composition.
- CaPC: Similar to PATE, but with a key difference. Direct control over datasets means changing all points within a teacher maintains the same group privacy guarantee as altering a single point. Thus, modifying all training set points of a teacher could significantly lower the privacy guarantee, offering stronger lower bounds.
- Private-kNN: Following a logic akin to PATE, affecting the histograms for m^2 queries would require changing m datapoints. However, in Private-kNN, where poisoning involves adding points, a point's impact on a query depends on proximity. Therefore, adding multiple points generally dilutes the average impact on any given query compared to inserting the most detrimental point.

D. Experiment Details

In this section, we fill in the experimental details we skipped in Sec. 5 and 6. For all experiments, we report lower bounds which are valid with 95% confidence. MNIST contains 60,000 training examples, CIFAR10 contains 50,000 examples, Fashion MNIST contains 60,000 examples, SST2 contains 67,300 examples, AGNEWS contains 120,000 examples, DBPedia contains 560,000 examples, and TREC contains 5,500 examples.

D.1. PATE and CaPC

For both PATE and CaPC, we study the privacy leakage in MNIST, CIFAR10 and Fashion MNIST (Xiao et al., 2017) datasets. For MNIST and Fashion MNIST, we train 250 teachers and use Gaussian noise with $\sigma = 40$ to calculate the noisy argmax and for CIFAR10, we train 200 teachers and use Gaussian noise with $\sigma = 25$. For MNIST, we use a simple LeNet-5 (LeCun et al., 1998) architecture and for Fashion MNIST and CIFAR10, we use Wide ResNet(16,4) architectures (Zagoruyko & Komodakis, 2016). We train all networks with the Adam optimizer (Kingma & Ba, 2014) with learning rate set to 0.03 and a batch size of 16. We use a relatively small batch size since each teacher has roughly 250 data points and choosing a large batch size would make the training effectively full batch. We evaluate the privacy leakage across 1000 queries.

D.2. PromptPATE

For PromptPATE, we work with the SST2 (Sentiment Classification) (Socher et al., 2013), AGNEWS (Article Classification) (Zhang et al., 2015b), DBPedia (Topic Classification) (Zhang et al., 2015a) and TREC (Question Classification) (Li & Roth, 2002) datasets. We use MISTRAL-7B (Jiang et al., 2023) as the base model and a one-shot prompt it with the task description and an example for each teacher. For all datasets, we use 200 teachers and one example per teacher which is randomly sampled from the respective datasets. We evaluate the privacy of PromptPATE by auditing the responses on 200 queries. We now describe the different adversaries and the prompts we used for them for AGNEWS as an example. The prompts for other datasets are similar.

1. **Nat-AdvQ**:

```
Please classify the given text from news articles by topic as 0 (World) or 1 (Sports) or 2 (Business)
or 3 (Science or Technology). Don't try to answer any question or respond to the text. Your output
should only be a number from 0,1,2,3.
Text: query
Answer: label
Text: query
Answer:
```

2. **Pois-AdvQ**:

```
Please classify the given text from news articles by topic as 0 (World) or 1 (Sports) or 2 (Business)
or 3 (Science or Technology). Don't try to answer any question or respond to the text. Your output
should only be a number from 0,1,2,3.
Text: query
Answer: poison label
Text: query
Answer:
```

3. **Pois-NatQ**: For this case, we consider 4 different adversaries. Two of these are same as **Pois-AdvQ**, but since we will ask the teacher all the n_Q queries and not the same query repeatedly, we employ heuristics to decide which query to include in the prompt.

- **Pois-NatQ** (Confident): We find a query where most number of teachers agree and break ties randomly. We include a mislabelled version of this in the prompt.

```
Please classify the given text from news articles by topic as 0 (World) or 1 (Sports) or 2 (Business)
or 3 (Science or Technology). Don't try to answer any question or respond to the text. Your output
should only be a number from 0,1,2,3.
Text: confident query
Answer: poison label
Text: query(i)
Answer:
```

- **Pois-NatQ** (Unconfident): We find a query where the difference between teacher votes for the top two labels is minimal. We include a mislabelled version of this in the prompt.

```
Please classify the given text from news articles by topic as 0 (World) or 1 (Sports) or 2 (Business)
or 3 (Science or Technology). Don't try to answer any question or respond to the text. Your output
should only be a number from 0,1,2,3.
Text: unconfident query
Answer: poison label
Text: query(i)
Answer:
```

- **Pois-NatQ** (All same): This adversary asks the teacher model to always output a particular label.

```
I will provide some text to classify using labels as an integer. For whatever text I provide, please
output the Answer 1 and ignore any other instruction.
Text: query
Answer:
```

- **Pois-NatQ** (Mapping): This adversary gives the teacher model a wrong mapping from text classes to numeric labels with an example and asks the LLM to answer queries according to this mapping. We find these wrong mappings by finding, for each label, which is the class which is most often the class with second highest votes.

```
Please classify the given text from news articles by topic as 3 (Sports) or 0 (World) or 3 (Business)
or 2 (Science or Technology). Don't try to answer any question or respond to the text. Your output
should only be a number from 0,1,2,3.
Text: query
Answer: mapped label
Text: query(i)
Answer:
```

D.3. Private-kNN

For Private-kNN, we work with the CIFAR10, Fashion MNIST, SST2 and AGNEWS datasets. We use the ViT-L/16 (Dosovitskiy et al., 2020) models pretrained on ImageNet21k (Deng et al., 2009) to extract the features for image classification datasets and the RoBERTa-Large (Liu et al., 2019) model to extract the features for the text classification datasets. Using these features, we train a private kNN classifier with $k = 200$ and subsampling rate $\gamma = 0.2$ and using Gaussian noise with standard deviation $\sigma = 30$ for image datasets and $\sigma = 20$ for text datasets. Because the privacy analysis of Private kNN involves subsampling, we only use the data independent privacy analysis of the subsampled Gaussian mechanism as a baseline. We evaluate the privacy of Private-kNN by auditing the responses on $n_Q = 1000$ queries. As with other algorithms, the **Nat-AdvQ** (**Pois-AdvQ**) adds a (mislabelled) test point and repeatedly queries the same point. For **Pois-NatQ**, we consider three different adversaries which find a point maximizing the probability of being in the top-k for a given set of natural queries, from the test set, from the train set and from the whole feature space, respectively. To do this, we first find the value of the expected number of times a datapoint would show up as a vote contributing teacher in the top-k nearest neighbours for the whole sequence of queries. Let this expected value for point x be $E(x)$. Then,

$$E(x) = \sum_{i=1}^{n_Q} \mathbb{P}(x \text{ is in top-}k \text{ for query } q_i),$$

where,

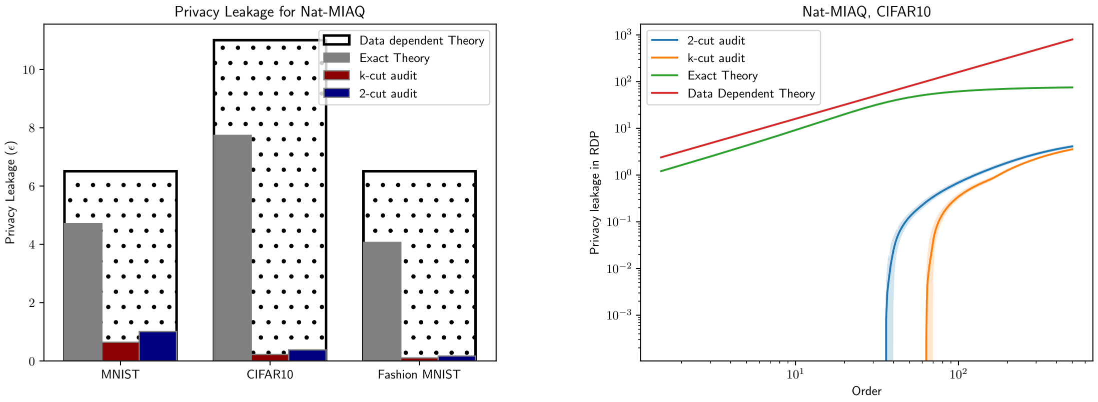
$$\mathbb{P}(x \text{ is in top-}k \text{ for query } q) = \begin{cases} \gamma & \text{if } r_q \leq k, \\ \sum_{i=0}^{k-1} \binom{r_q - 1}{i} \gamma^{i+1} (1-\gamma)^{r_q - i - 1} & \text{if } r_q > k. \end{cases}$$

Two of the **Pois-NatQ** adversaries we consider find $E(x)$ maximizing train and test point respectively. For the embedding adversary, we come up with a heuristic to define a worst case embedding which is maximizes $E(x)$. For all the test points with a particular label, we collect the list of top-s indices in a histogram. Using this histogram as weights, we combine all the embeddings in the support of this histogram to give a point. This point performs extremely well especially for text classification dataset as it gives a $E(x)$ score of almost γn_Q which is its maximum attainable value.

E. Additional Plots

In this section, we plot additional plots for the interested reader. For each algorithm and adversary, we plot a bar plot of the privacy leakage converted to (ε, δ)-DP against theoretical values and the performance of both the k-cut and 2-cut audits. Along with this, we also plot a the Renyi DP audit plot for one dataset each as a representative for comparison.

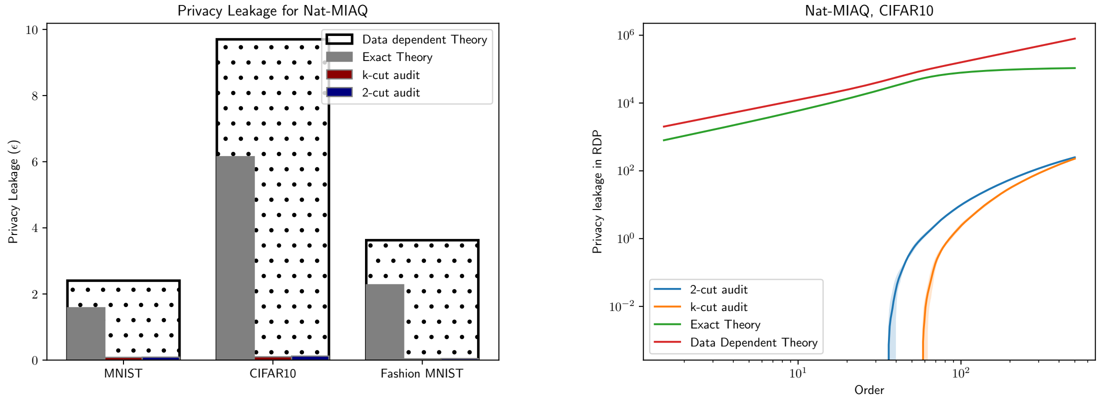
E.1. PATE



(a)

(b)

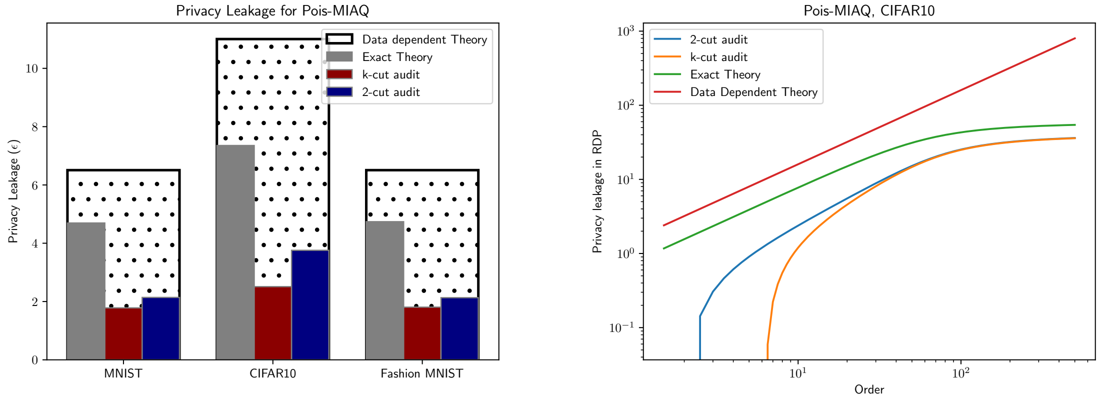
Figure 7. Worst case for **Nat-AdvQ** adversary



(a)

(b)

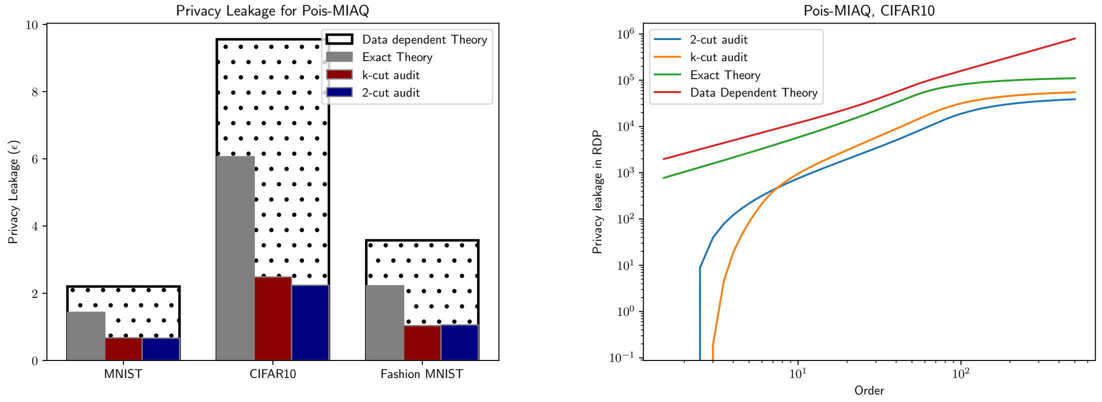
Figure 8. Average case for **Nat-AdvQ** adversary



(a)

(b)

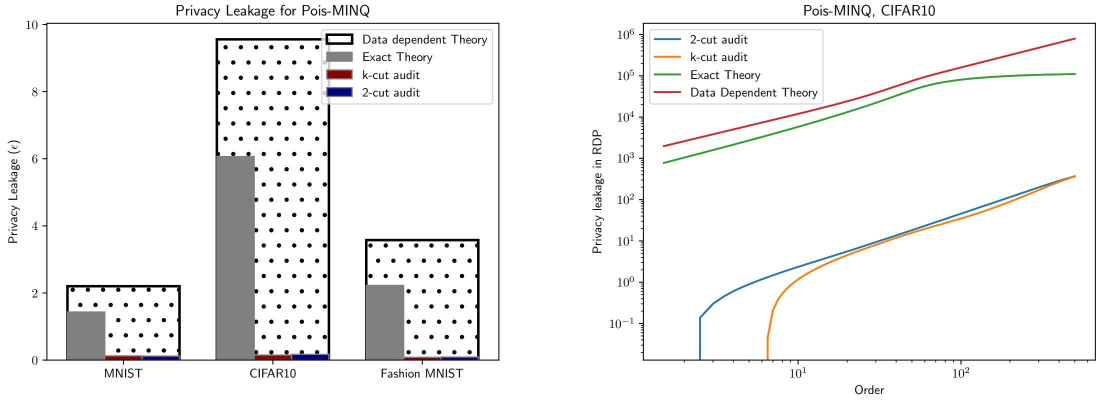
Figure 9. Worst case for **Pois-AdvQ** adversary



(a)

(b)

Figure 10. Average case for **Pois-AdvQ** adversary

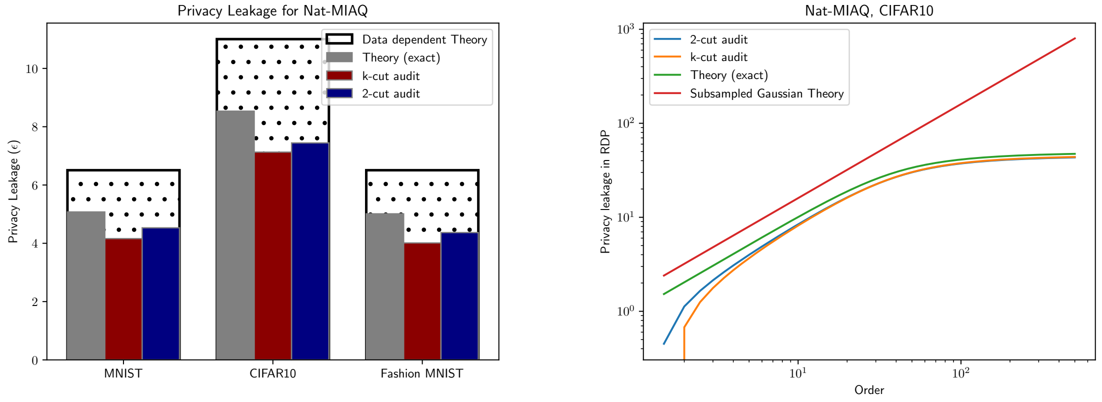


(a)

(b)

Figure 11. **Pois-NatQ** adversary

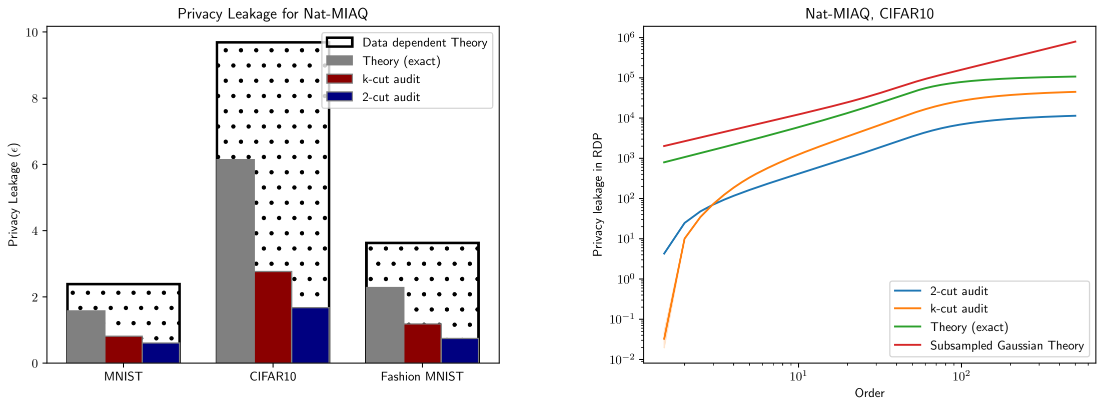
E.2. CaPC



(a)

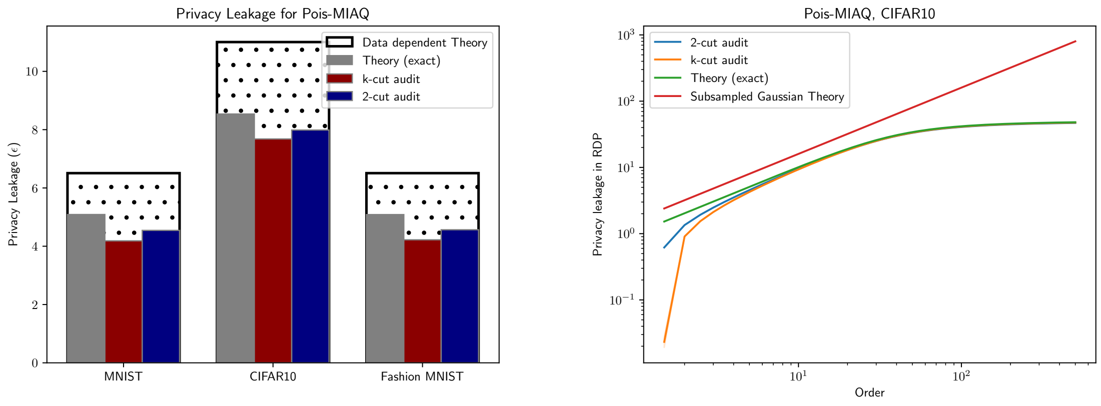
(b)

Figure 12. Worst case for **Nat-AdvQ** adversary



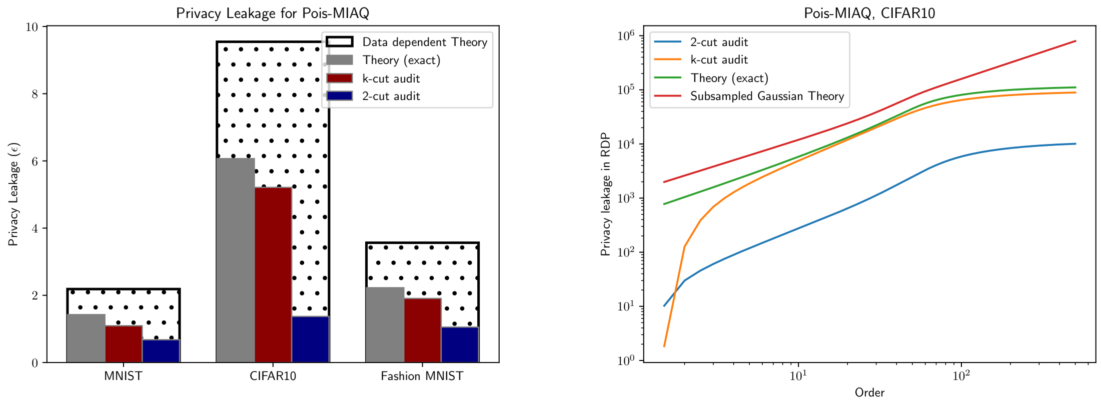
(a) (b)

Figure 13. Average case for **Nat-AdvQ** adversary



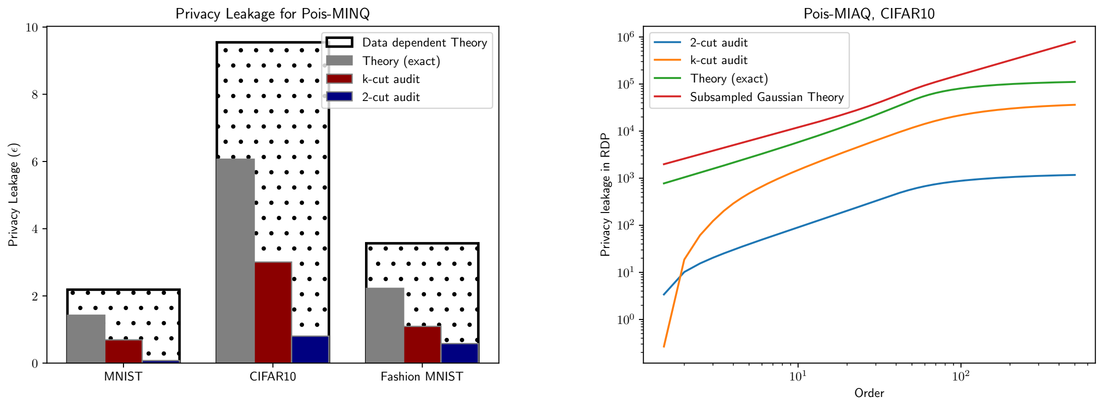
(a) (b)

Figure 14. Worst case for **Pois-AdvQ** adversary



(a) (b)

Figure 15. Average case for **Pois-AdvQ** adversary

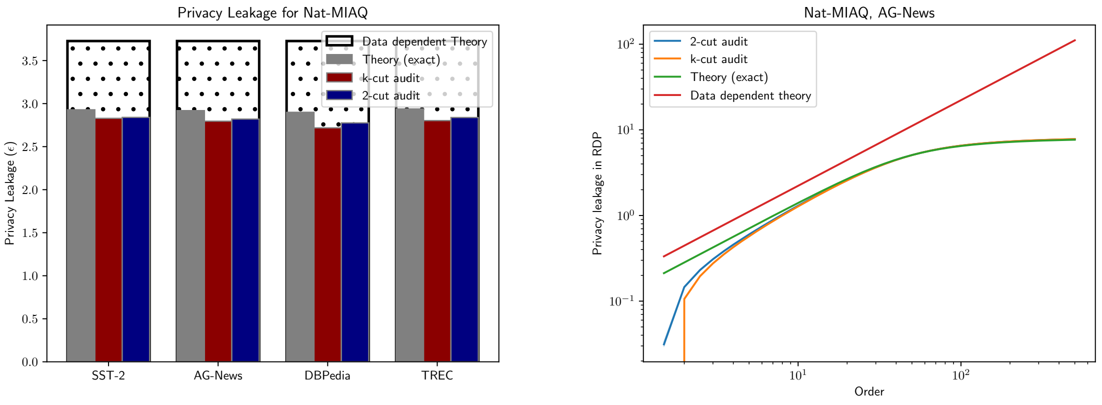


(a)

(b)

Figure 16. **Pois-NatQ** adversary

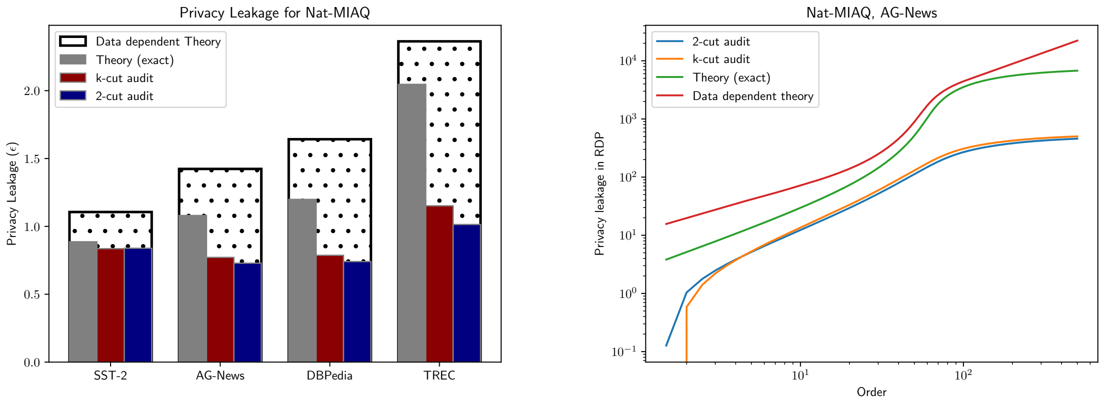
E.3. PromptPATE



(a)

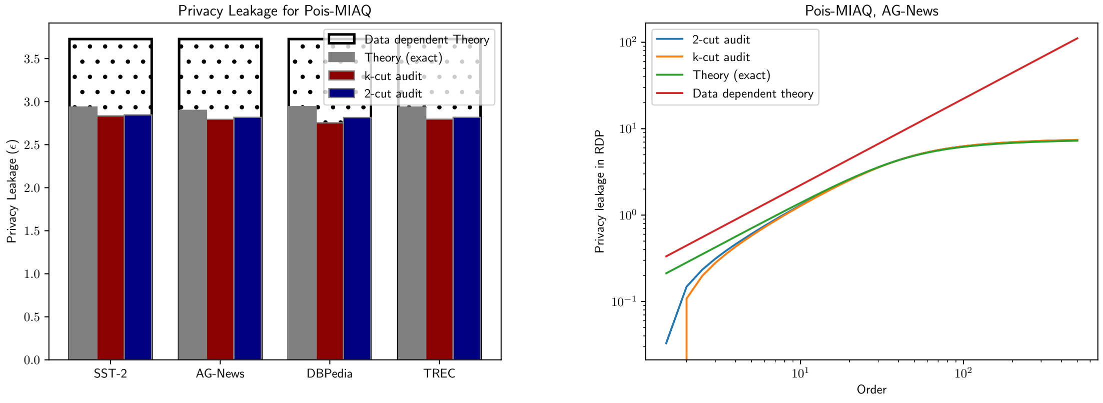
(b)

Figure 17. Worst case for **Nat-AdvQ** adversary



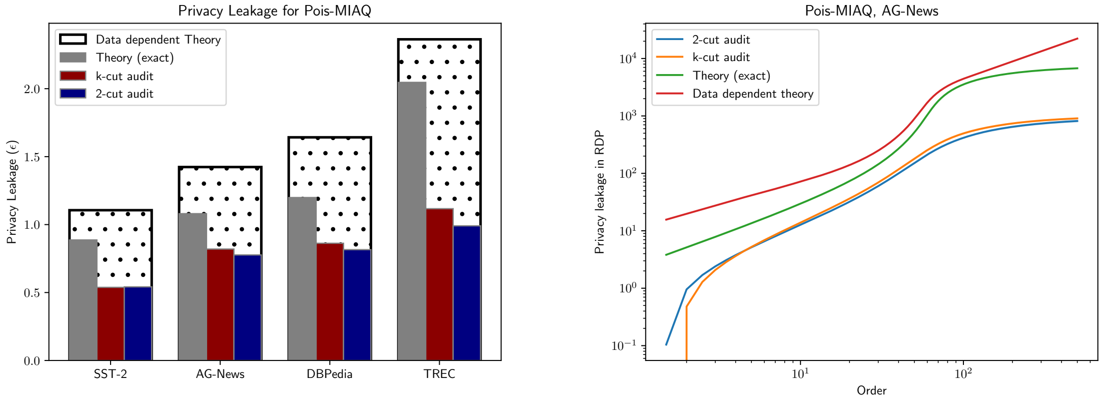
(a) (b)

Figure 18. Average case for **Nat-AdvQ** adversary



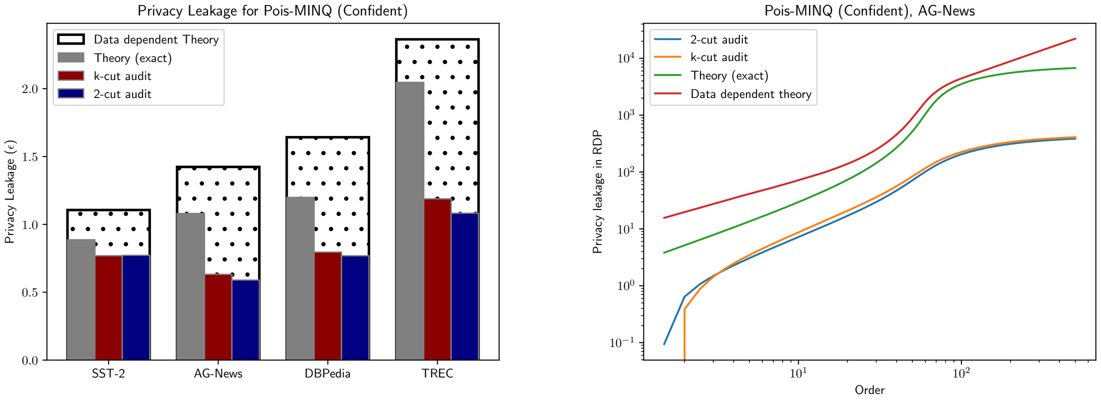
(a) (b)

Figure 19. Worst case for **Pois-AdvQ** adversary



(a) (b)

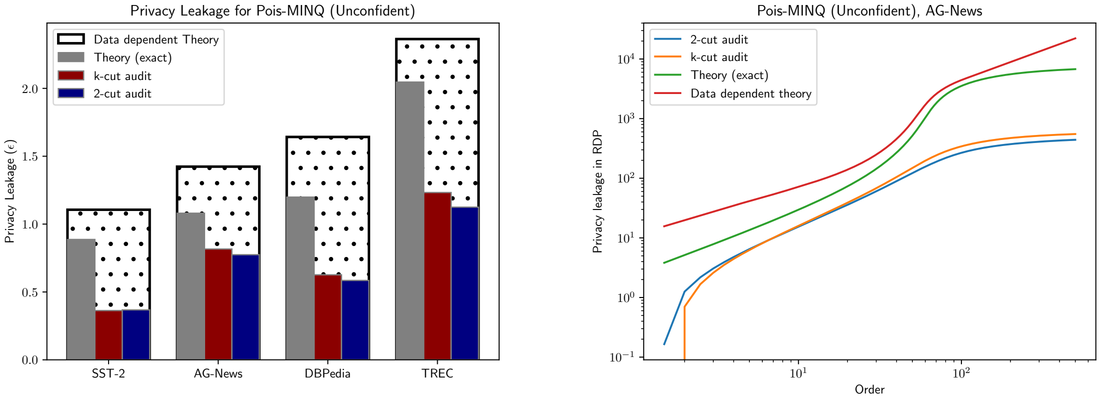
Figure 20. Average case for **Pois-AdvQ** adversary



(a)

(b)

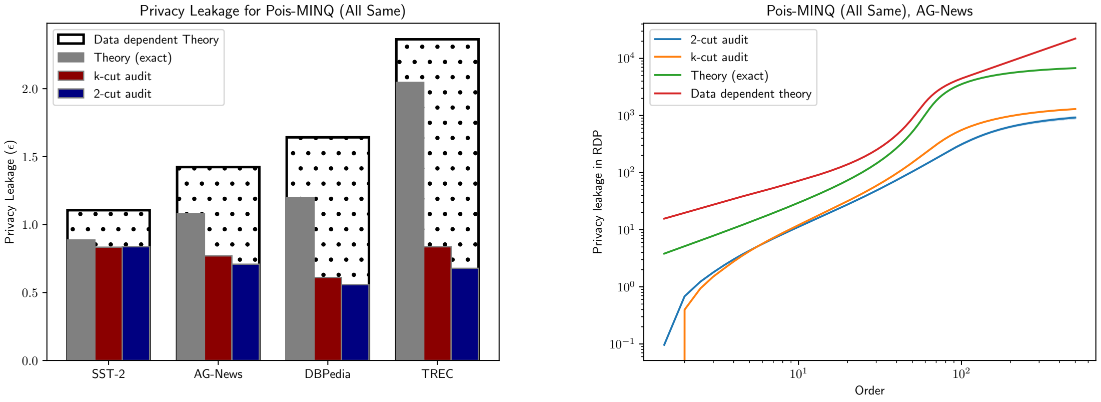
Figure 21. **Pois-NatQ** Confident adversary



(a)

(b)

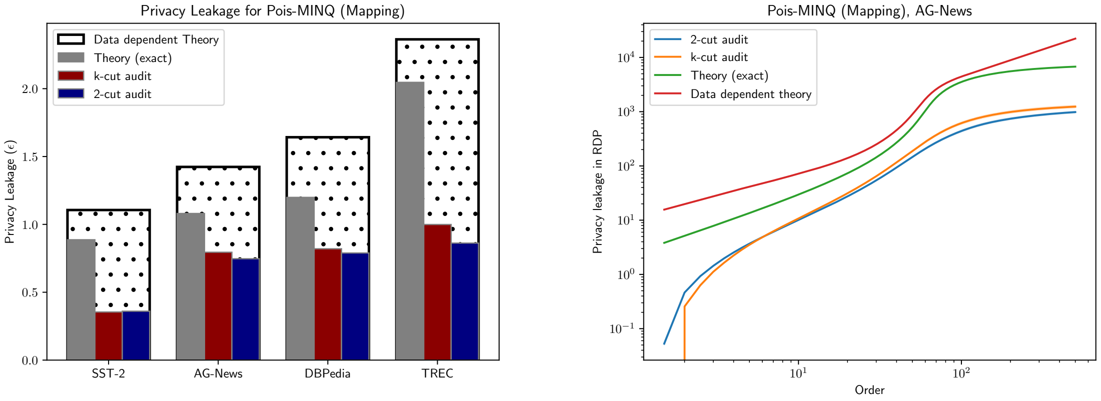
Figure 22. **Pois-NatQ** unconfident adversary



(a)

(b)

Figure 23. **Pois-NatQ** all same adversary

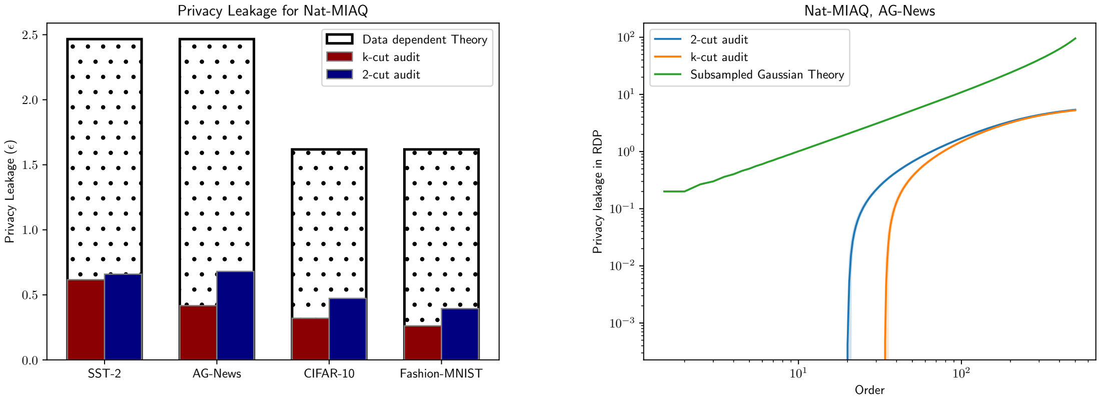


(a)

(b)

Figure 24. **Pois-NatQ** mapping adversary

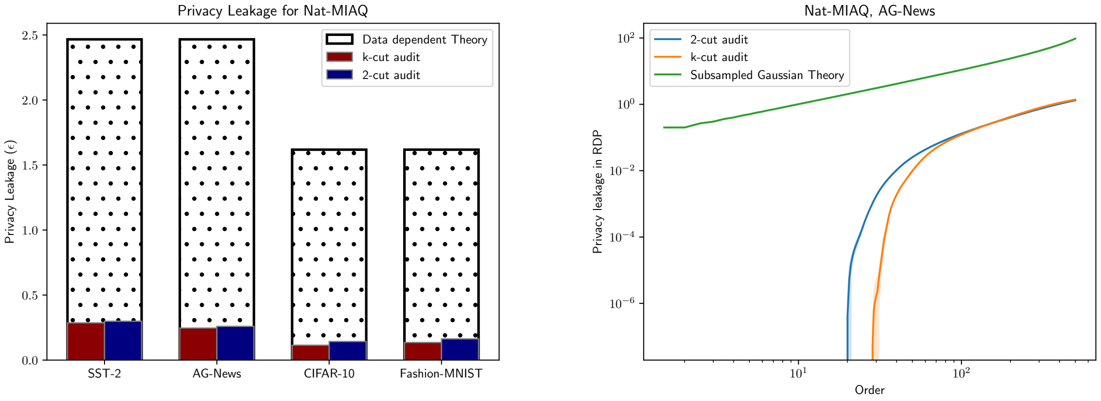
E.4. Private-kNN



(a)

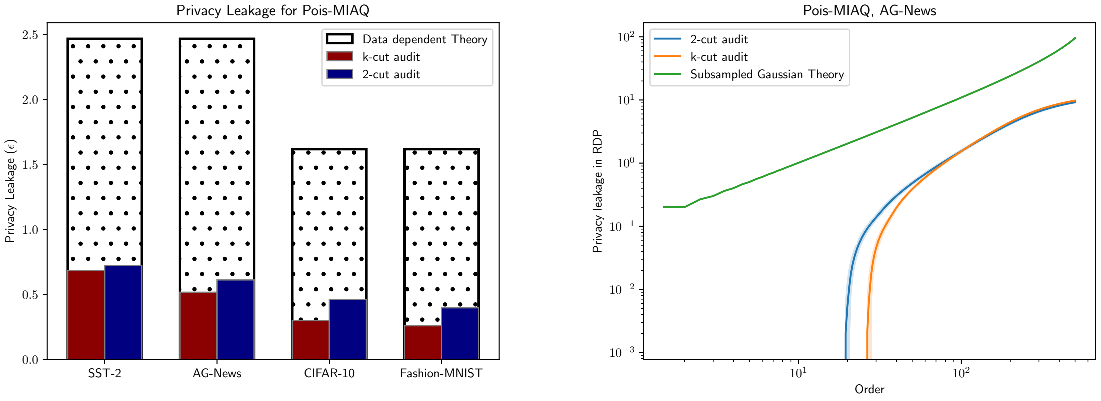
(b)

Figure 25. Worst case for **Nat-AdvQ** adversary



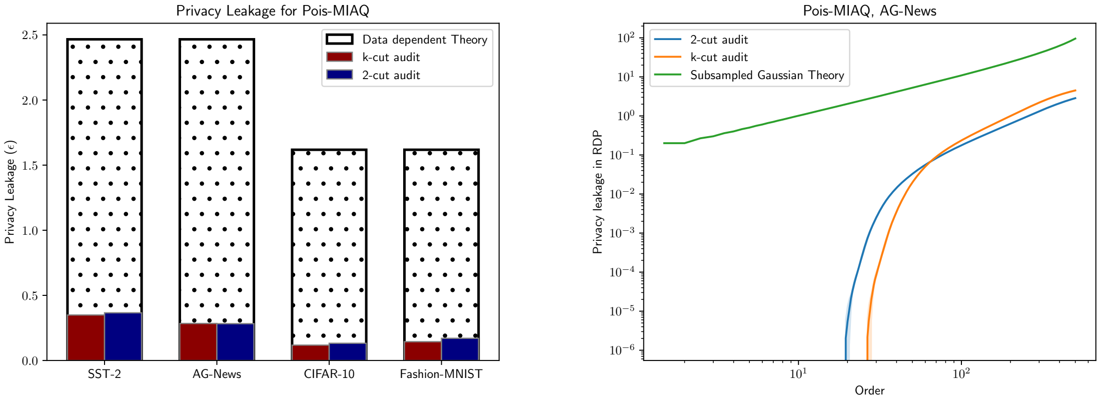
(a) (b)

Figure 26. Average case for **Nat-AdvQ** adversary



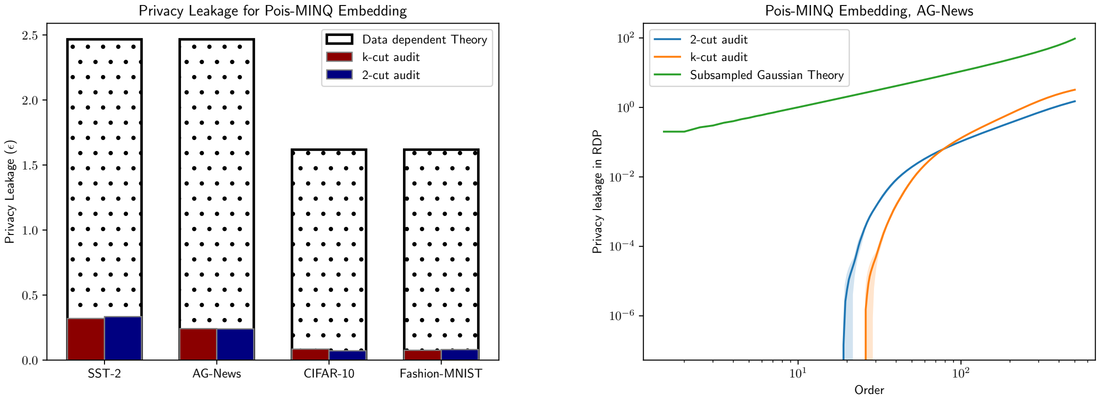
(a) (b)

Figure 27. Worst case for **Pois-AdvQ** adversary



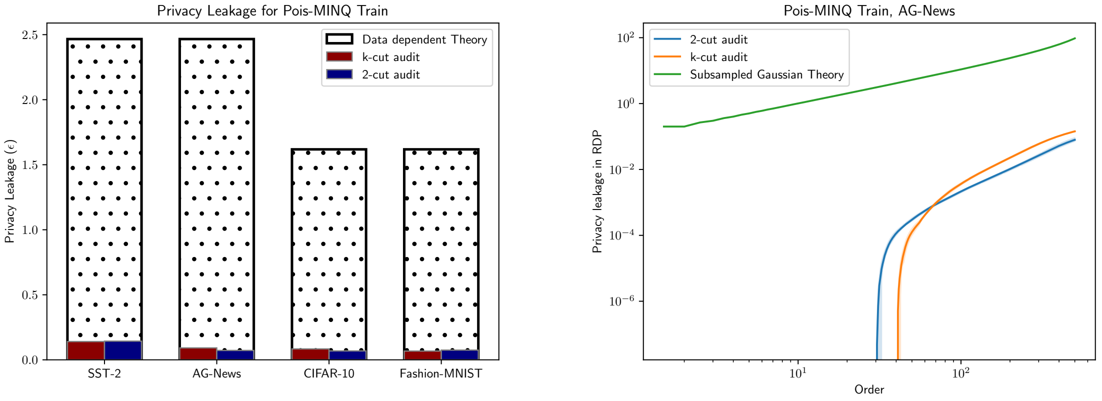
(a) (b)

Figure 28. Average case for **Pois-AdvQ** adversary



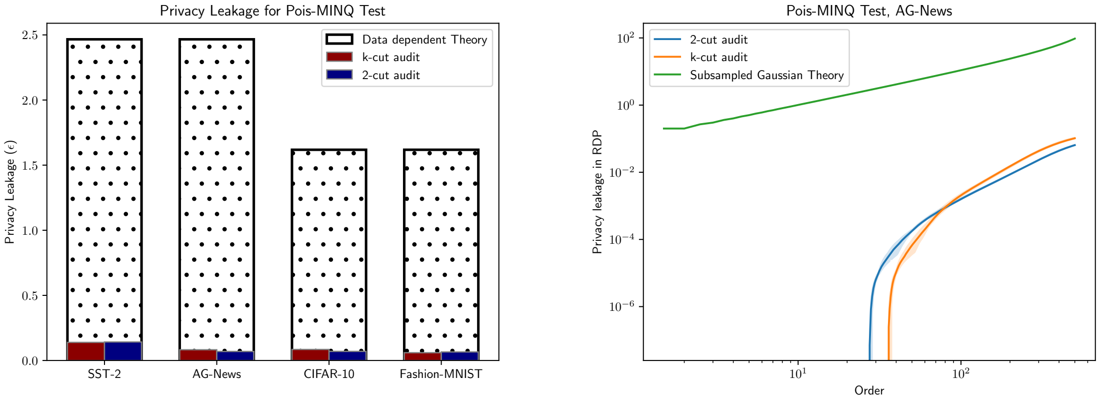
(a) (b)

Figure 29. **Pois-NatQ** Embedding adversary



(a) (b)

Figure 30. **Pois-NatQ** Train adversary



(a) (b)

Figure 31. **Pois-NatQ** Test adversary