The Last Iterate Advantage: Empirical Auditing and Principled Heuristic Analysis of Differentially Private SGD

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a simple heuristic privacy analysis of noisy clipped stochastic gradient descent (DP-SGD) in the setting where only the last iterate is released and the intermediate iterates remain hidden. Namely, our heuristic assumes a linear structure for the model.

We show experimentally that our heuristic is predictive of the outcome of privacy auditing applied to various training procedures. Thus it can be used prior to training as a rough estimate of the final privacy leakage. We also probe the limitations of our heuristic by providing some artificial counterexamples where it underestimates the privacy leakage.

The standard composition-based privacy analysis of DP-SGD effectively assumes that the adversary has access to all intermediate iterates, which is often unrealistic. However, this analysis remains the state of the art in practice. While our heuristic does not replace a rigorous privacy analysis, it illustrates the large gap between the best theoretical upper bounds and the privacy auditing lower bounds and sets a target for further work to improve the theoretical privacy analyses.

1 Introduction

Differential privacy (DP) [DMNS06] defines a measure of how much private information from the training data leaks through the output of an algorithm. The standard differentially private algorithm for deep learning is DP-SGD [BST14; ACGMMTZ16], which differs from ordinary stochastic gradient descent in two ways: the gradient of each example is clipped to bound its norm and then Gaussian noise is added at each iteration.

The standard privacy analysis of DP-SGD is based on composition [BST14; ACGMMTZ16; Mir17; Ste22; KJH20]. In particular, it applies to the setting where the privacy adversary has access to all intermediate iterates of the training procedure. In this setting, the analysis is known to be tight [NSTPC21; NHSBTJCT23]. However, in practice, potential adversaries rarely have access to the intermediate iterates of the training procedure, rather they only have access to the final model. Access to the final model can either be through queries to an API or via the raw model weights. The key question motivating our work is the following.

> Is it possible to obtain sharper privacy guarantees for DP-SGD when the adversary only has access to the final model, rather than all intermediate iterates?

1.1 Background & Related Work

The question above has been studied from two angles: Theoretical upper bounds, and privacy auditing lower bounds. Our goal is to shed light on this question from a third angle via principled heuristics.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

A handful of theoretical analyses [FMTT18; CYS21; YS22; AT22; BSA24] have shown that asymptotically the privacy guarantee of the last iterate of DP-SGD can be far better than the standard composition-based analysis that applies to releasing all iterates. In particular, as the number of iterations increases, these analyses give a privacy guarantee that converges to a constant (depending on the loss function and the scale of the noise), whereas the standard composition-based analysis would give a privacy guarantee that increases forever. Unfortunately, these theoretical analyses are only applicable under strong assumptions on the loss function, such as (strong) convexity and smoothness. We lack an understanding of how well they reflect the "real" privacy leakage.

Privacy auditing [JUO20; DWWZK18; BGDCTV18; SNJ23; TTSSJC22; ZBWTSRPNK22] complements theoretical analysis by giving empirical lower bounds on the privacy leakage. Privacy auditing works by performing a membership inference attack [SSSS17; HSRDTMPSNC08; SOJH09; DSSUV15]. That is, it constructs neighbouring inputs and demonstrates that the corresponding output distributions can be distinguished well enough to imply a lower bound on the differential privacy parameters. In practice, the theoretical privacy analysis may give uncomfortably large values for the privacy leakage (e.g., $\varepsilon > 10$); in this case, privacy auditing may be used as evidence that the "real" privacy leakage is lower. There are settings where the theoretical analysis is matched by auditing, such as when all intermediate results are released [NSTPC21; NHSBTJCT23]. However, despite significant work on privacy auditing and membership inference [CCNSTT22; BTRKMW24; WBKBGGG23; LF20; SDSOJ19; ZLS23], a large gap remains between the theoretical upper bounds and the auditing lower bounds [AKOOMS23; NHSBTJCT23] when only the final parameters are released. This observed gap is the starting point for our work.

1.2 Our Contributions

We propose a *heuristic* privacy analysis of DP-SGD in the setting where only the final iterate is released. Our experiments demonstrate that this heuristic analysis consistently provides an upper bound on the privacy leakage measured by privacy auditing tools in realistic deep learning settings.

Our heuristic analysis corresponds to a worst-case theoretical analysis under the assumption that the loss functions are linear. This case is simple enough to allow for an exact privacy analysis whose parameters are can be computed numerically (Theorem 1). Our consideration of linear losses is built on the observation that current auditing techniques achieve the highest ε values when the gradients of the canaries – that is, the examples that are included or excluded to test the privacy leakage – are fixed and independent from the gradients of the other examples. This is definitely the case for linear losses; the linear assumption thus allows us to capture the setting where current attacks are most effective. Linear loss functions are also known to be the worst case for the non-subsampled (i.e., full batch) case; see Appendix B. Assuming linearity is unnatural from an optimization perspective, as there is no minimizer. But, from a privacy perspective, we show that it captures the state of the art.

We also probe the limitations of our heuristic and give some artificial counterexamples where it underestimates empirical privacy leakage. One class of counterexamples exploits the presence of a regularizer. Roughly, the regularizer partially zeros out the noise that is added for privacy. However, the regularizer also partially zeros out the signal of the canary gradient. These two effects are almost balanced, which makes the counterexample very delicate. In a second class of counterexamples, the data is carefully engineered so that the final iterate effectively encodes the entire trajectory, in which case there is no difference between releasing the last iterate and all iterates.

Implications: Heuristics cannot replace rigorous theoretical analyses. However, our heuristic can serve as a target for future improvements to both privacy auditing as well as theoretical analysis. For privacy auditing, matching or exceeding our heuristic is a more reachable goal than matching the theoretical upper bounds, although our experimental results show that even this would require new attacks. When theoretical analyses fail to match our heuristic, we should identify why there is a gap, which builds intuition and could point towards further improvements.

Given that privacy auditing is computationally intensive and difficult to perform correctly [AZT24], we believe that our heuristic can also be valuable in practice. In particular, our heuristic can be used prior to training (e.g., during hyperparameter selection) to predict the outcome of privacy auditing when applied to the final model. (This is a similar use case to scaling laws.)

2 Linearized Heuristic Privacy Analysis

Theorem 1 presents our heuristic differential privacy analysis of DP-SGD (which we present in Algorithm 1 for completeness; note that we include a regularizer r whose gradient is *not* clipped, because it does not depend on the private data $\mathbf{x}$). We consider Poisson subsampled minibatches and add/remove neighbours, as is standard in the differential privacy literature.

Algorithm 1 Noisy Clipped Stochastic Gradient Descent (DP-SGD) [BST14; ACGMMTZ16]

function DP-SGD($\mathbf{x} \in \mathcal{X}^n$, $T \in \mathbb{N}$, $q \in [0,1]$, $\eta \in (0,\infty)$, $\sigma \in (0,\infty)$, $\ell : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}$, $r : \mathbb{R}^d \to \mathbb{R}$)
  Initialize model $\mathbf{m}_0 \in \mathbb{R}^d$.
  for $t = 1 \cdots T$ **do**
    Sample minibatch $B_t \subseteq [n]$ including each element independently with probability q.
    Compute gradients of the loss $\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)$ for all $i \in B_t$ and of the regularizer $\nabla_{\mathbf{m}_{t-1}} r(\mathbf{m}_{t-1})$.
    Clip loss gradients: $\mathsf{clip}\left(\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)\right) := \frac{\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)}{\max\{1, \|\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)\|_2\}}$.
    Sample noise $\xi_t \leftarrow \mathcal{N}(0, \sigma^2 I_d)$.
    Update

$$\mathbf{m}_t = \mathbf{m}_{t-1} - \eta \cdot \left(\begin{array}{c} \sum_{i \in B_t} \mathsf{clip}\left(\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)\right) \\ + \nabla_{\mathbf{m}_{t-1}} r(\mathbf{m}_{t-1}) + \xi_t \end{array} \right).$$

  end for
  if `last_iterate_only` **then**
    return $\mathbf{m}_T$
  else if `intermediate_iterates` **then**
    return $\mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{T-1}, \mathbf{m}_T$
  end if
end function

Our analysis takes the form of a conditional privacy guarantee. Namely, under the assumption that the loss and regularizer are linear, we obtain a fully rigorous differential privacy guarantee. The heuristic is to apply this guarantee to loss functions that are not linear (such as those that arise in deep learning applications). Our thesis is that, in most cases, the conclusion of the theorem is still a good approximation, even when the assumption does not hold.

Recall that a function $\ell : \mathbb{R}^d \to \mathbb{R}$ is linear if there exist $\alpha \in \mathbb{R}^d$ and $\beta \in \mathbb{R}$ such that $\ell(\mathbf{m}) = \langle \alpha, \mathbf{m} \rangle + \beta$ for all $\mathbf{m}$.

Theorem 1 (Privacy of DP-SGD for linear losses)**.** *Let $\mathbf{x}, T, q, \eta, \sigma, \ell, r$ be as in Algorithm 1. Assume r and $\ell(\cdot, x)$, for every $x \in \mathcal{X}$, are linear.*

Letting

$$P := \mathsf{Binomial}(T, q) + \mathcal{N}(0, \sigma^2 T), Q := \mathcal{N}(0, \sigma^2 T), \tag{1}$$

DP-SGD with `last_iterate_only` satisfies (ε, δ)-differential privacy with $\varepsilon \geq 0$ arbitrary and

$$\delta = \delta_{T,q,\sigma}(\varepsilon) := \max\{H_{e^\varepsilon}(P, Q), H_{e^\varepsilon}(Q, P)\}. \tag{2}$$

Here, H_{e^ε} denotes the e^ε-hockey-stick-divergence $H_{e^\varepsilon}(P, Q) := \sup_S P(S) - e^\varepsilon Q(S)$.

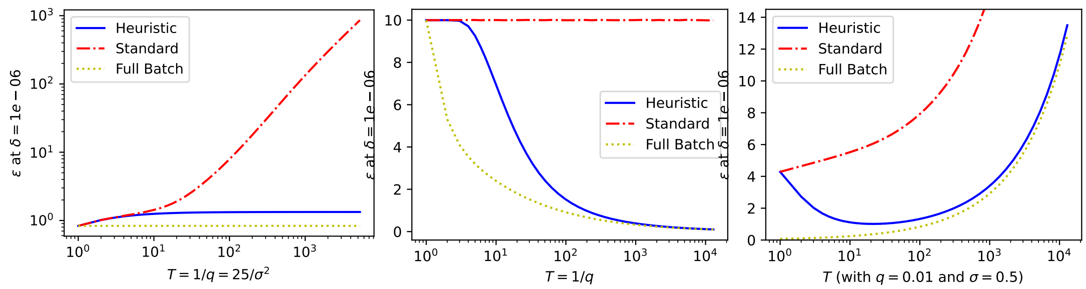
Equation 1 gives us a value of the privacy failure probability parameter δ. But it is more natural to work with the privacy loss bound parameter ε, which can be computed by inverting the formula:

$$\varepsilon_{T,q,\sigma}(\delta) := \min\{\varepsilon \geq 0 : \delta_{T,q,\sigma}(\varepsilon) \leq \delta\}. \tag{3}$$

Both $\delta_{T,q,\sigma}(\varepsilon)$ and $\varepsilon_{T,q,\sigma}(\delta)$ can be computed using existing open-source DP accounting libraries [Goo20]. We also provide a self-contained & efficient method for computing them in Appendix A.

The proof of Theorem 1 is deferred to Appendix A, but we sketch the main ideas: Under the linearity assumption, the output of DP-SGD is just a sum of the gradients and noises. We can reduce to dimension $d = 1$, since the only relevant direction is that of the gradient of the canary[1] (which is constant). We can also ignore the gradients of the other examples. Thus, by rescaling, the worst case pair of output distributions can be represented as in Equation 1. Namely, $Q = \sum_{t=1}^{T} \xi_t$ is simply the noise $\xi_t \leftarrow \mathcal{N}(0, \sigma^2)$ summed over T iterations; this corresponds to the case where the canary is excluded. When the canary is included, it is sampled with probability q in each iteration and thus the total number of times it is sampled over T iterations is $\mathsf{Binomial}(T, q)$. Thus P is the sum of the contributions of the canary and the noise. Finally the definition of differential privacy lets us compute ε and δ from this pair of distributions. Tightness follows from the fact that there exists a loss function and pair of inputs such that the corresponding outputs of DP-SGD matches the pair P and Q.

[1] *The canary* refers to the individual datapoint that is added or removed between neighbouring datasets. This terminology is used in the privacy auditing/attacks literature inspired on the expression "canary in a coalmine."



(a) Keep Tp and $T\sigma^2$ constant. (b) Keep Standard DP constant. (c) Keep p and σ constant.

Figure 1: Comparison of our heuristic to baselines in various parameter regimes. Horizontal axis is number of iterations T and vertical axis is ε such that we have $(\varepsilon, 10^{-6})$-DP.

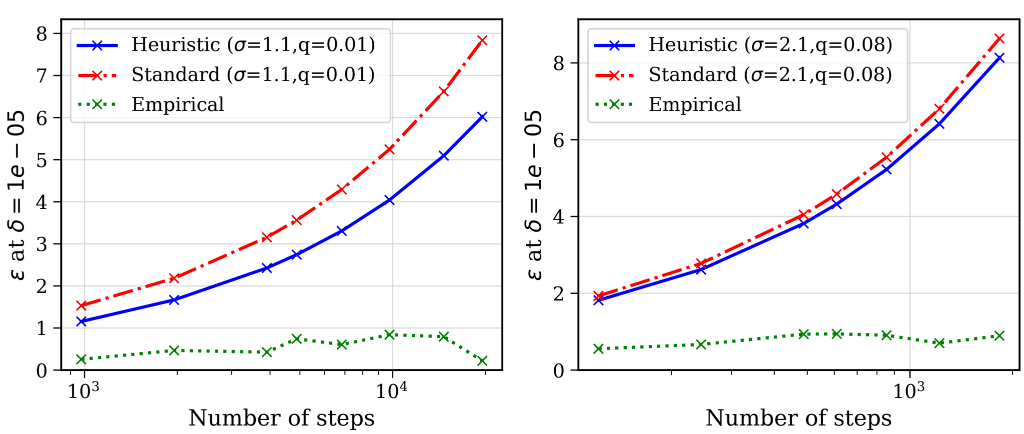
2.1 Baselines

In addition to privacy auditing, we compare our heuristic to two different baselines in Figure 1. The first is the standard, composition-based analysis. We use the open-source library from Google [Goo20], which computes a tight DP guarantee for DP-SGD with `intermediate_iterates`. Because DP-SGD with `intermediate_iterates` gives the adversary more information than with `last_iterate_only`, this will always give at least as large an estimate for ε as our heuristic.

We also consider approximating DP-SGD by full batch DP-GD. That is, set $q = 1$ and rescale the learning rate η and noise multiplier σ to keep the expected step and privacy noise variance constant:

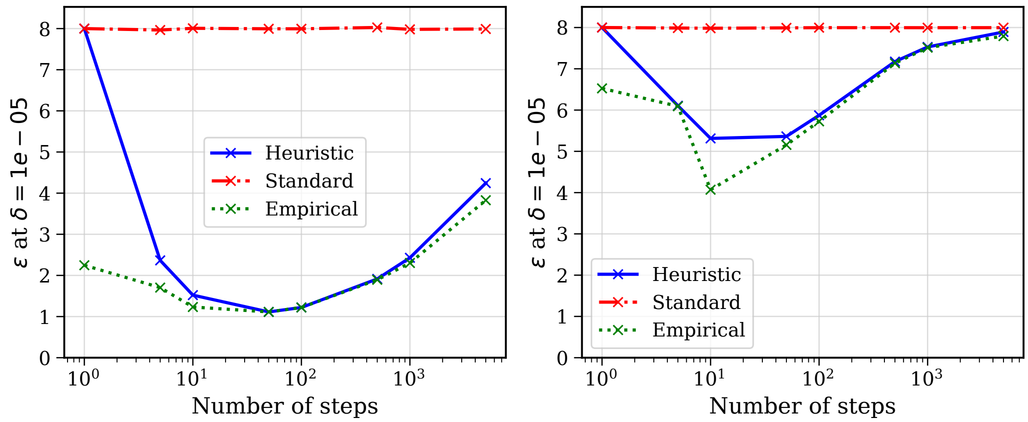
$$\underbrace{\text{DP-SGD}(\mathbf{x}, T, q, \eta, \sigma, \ell, r)}_{\text{batch size } \approx nq,\ T \text{ iterations},\ Tq \text{ epochs}} \approx \underbrace{\text{DP-SGD}(\mathbf{x}, T, 1, \eta \cdot q, \sigma/q, \ell, r)}_{\text{batch size } n,\ T \text{ iterations},\ T \text{ epochs}}. \quad (4)$$

The latter algorithm is full batch DP-GD since at each step it includes each data point in the batch with probability 1. Since full batch DP-GD does not rely on privacy amplification by subsampling, it is much easier to analyze its privacy guarantees. Interestingly, there is no difference between full batch DP-GD with `last_iterate_only` and with `intermediate_iterates`; see Appendix B. Full batch DP-GD generally has better privacy guarantees than the corresponding minibatch DP-SGD and so this baseline usually (but not always) gives smaller values for the privacy leakage ε than our heuristic. In practice, full batch DP-GD is too computationally expensive to run. But we can use it as an idealized comparison point for the privacy analysis.



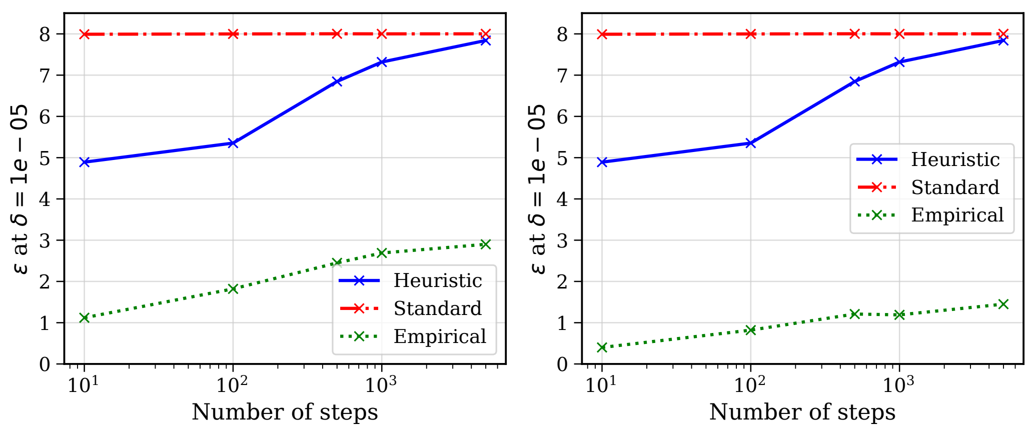
(a) $q = 0.01$ (b) $q = 0.08$

Figure 2: Black-box gradient space attacks fail to achieve tight auditing when other data points are sampled from the data distribution. Heuristic and standard bounds diverge from empirical results, indicating the attack's ineffectiveness. This contrasts with previous work which tightly auditing with access to intermediate updates.



(a) $q = 0.01$ (b) $q = 0.1$

Figure 3: For gradient space attacks with adversarial datasets, the empirical epsilon (ε) closely tracks the final epsilon except for at small step counts, where distinguishing is more challenging. This is evident at both subsampling probability values we study ($q = 0.01$ and $q = 0.1$).



(a) All zero gradient inputs (b) CIFAR10 Dataset

Figure 4: Input space attacks show promising results with both natural and blank image settings, although blank images have higher attack success. These input space attacks achieve tighter results than gradient space attacks in the natural data setting, in contrast to findings from prior work.

3 Empirical Evaluation via Privacy Auditing

Setup: We follow the construction of Nasr, Song, Thakurta, Papernot, and Carlini [NSTPC21] where we have 3 entities, adversarial crafter, model trainer, and distinguisher. In this paper, we assume the distinguisher only has access the final iteration of the model parameters. We use the CIFAR10 dataset [Ale09] with a WideResNet model [ZK16] unless otherwise specified; in particular, we follow the training setup of De, Berrada, Hayes, Smith, and Balle [DBHSB22], where we train and audit a model with 79% test accuracy and, using the standard analysis, $(\varepsilon = 8, \delta = 10^{-5})$-DP. For each experiment we trained 512 CIFAR10 models with and without the canary (1024 total). To compute the empirical lower bounds we use the PLD approach with Clopper-Pearson confidence intervals used by Nasr, Hayes, Steinke, Balle, Tramèr, Jagielski, Carlini, and Terzis [NHSBTJCT23]. Here we assume the adversary knows the sampling rate and the number of iterations and is only estimating the noise multiplier used in DP-SGD, from which the reported privacy parameters (ε and δ) are derived.

3.1 Experimental Results

We implement state-of-the-art attacks from prior work [NSTPC21; NHSBTJCT23]. These attacks heavily rely on the intermediate steps and, as a result, do not achieve tight results. In the next section, we design specific attacks for our heuristic privacy analysis approach to further understand its limitations and potential vulnerabilities. We used Google Cloud `A2-megagpu-16g` machines with 16 Nvidia A100 40GB GPUs. Overall, we use roughly 33,000 GPU hours for our experiments.

Gradient Space Attack: The most powerful attacks in prior work are gradient space attacks where the adversary injects a malicious gradient directly into the training process, rather than an example; prior work has shown that this attack can produce tight lower bounds, independent of the dataset and model used for training [NHSBTJCT23]. However, these previous attacks require access to all intermediate training steps to achieve tight results. Here, we use canary gradients in two settings: one where the other data points are non-adversarial and sampled from the real training data, and another where the other data points are designed to have very small gradients (≈ 0). This last setting was shown by [NSTPC21] to result in tighter auditing. In all attacks, we assume the distinguisher has access to all adversarial gradient vectors. For malicious gradients, we use Dirac gradient canaries, where gradient vectors consist of zeros in all but a single index. In both cases, the distinguishing test measures the dot product of the final model checkpoint and the gradient canary.

Figure 2 summarizes the results for the non-adversarial data setting, with other examples sampled from the true training data. In this experiment, we fix noise magnitude and subsampling probability, and run for various numbers of training steps. While prior work has shown tight auditing in this setting, we find an adversary without access to intermediate updates obtains much weaker attacks. Indeed, auditing with this strong attack results even in much lower values than the heuristic outputs.

Our other setting assumes the other data points are maliciously chosen. We construct an adversarial "dataset" of $m + 1$ gradients, m of which are zero, and one gradient is constant (with norm equal to the clipping norm), applying gradients directly rather than using any examples. As this experiment does not require computing gradients, it is very cheap to run more trials, so we run this procedure $N = 100,000$ times with the gradient canary, and N times without it, and compute an empirical estimate for ε with these values. We plot the results of this experiment in Figure 3 together with the ε output by the theoretical analysis and the heuristic, fixing the subsampling probability and varying the number of update steps. We adjust the noise parameter to ensure the standard theoretical analysis produces a fixed ε bound. The empirical measured ε is close to the heuristic ε except for when training with very small step counts: we expect this looseness to be the result of statistical effects, as lower step counts have higher relative variance at a fixed number of trials.

Input Space Attack: In practice, adversaries typically cannot insert malicious gradients freely in training steps. Therefore, we also study cases where the adversary is limited to inserting malicious inputs into the training set. Label flip attacks are one of the most successful approaches used to audit DP machine learning models in prior work [NHSBTJCT23; SNJ23]. For input space attacks, we use the loss of the malicious input as a distinguisher. Similar to our gradient space attacks, we consider two settings for input space attacks: one where other data points are correctly sampled from the dataset, and another where the other data points are blank images.

Figure 4 summarizes the results for this setting. Comparing to Figure 2, input space attacks achieve tighter results than gradient space attacks. This finding is in stark contrast to prior work. The reason is that input space attacks do not rely on intermediate iterates, so they transfer well to our setting.

In all the cases discussed so far, the empirical results for both gradient and input attacks fall below the heuristic analysis and do not violate the upper bounds based on the underlying assumptions. This suggests that the heuristic might serve as a good indicator for assessing potential vulnerabilities. However, in the next section, we delve into specific attack scenarios that exploit the assumptions used in the heuristic analysis to create edge cases where the heuristic bounds are indeed violated.

4 Counterexamples

We now test the limits of our heuristic by constructing some artificial counterexamples. That is, we construct inputs to DP-SGD with `last_iterate_only` such that the true privacy loss exceeds the bound given by our heuristic. While we do not expect the contrived structures of these examples to manifest in realistic learning settings, they highlight the difficulties of formalizing settings where the heuristic gives a provable upper bound on the privacy loss.

4.1 Warmup: Zeroing Out The Model Weights

We begin by noting the counterintuitive fact that our heuristic $\varepsilon_{T,q,\sigma}(\delta)$ is *not* always monotone in the number of steps T when the other parameters σ, q, δ are kept constant. This is shown in Figure 1c. More steps means there is both more noise and more signal from the gradients; these effects partially cancel out, but the net effect can be non-monotone.

We can use a regularizer $r(\mathbf{m}) = \|\mathbf{m}\|_2^2/2\eta$ so that $\eta \cdot \nabla_{\mathbf{m}} r(\mathbf{m}) = \mathbf{m}$. This regularizer zeros out the model from the previous step, i.e., the update of DP-SGD becomes

$$\mathbf{m}_t = \mathbf{m}_{t-1} - \eta \cdot \left(\sum_{i \in B_t} \mathsf{clip}\left(\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)\right) + \nabla_{\mathbf{m}_{t-1}} r(\mathbf{m}_{t-1}) + \xi_t \right) \tag{5}$$

$$= \eta \cdot \sum_{i \in B_t} \mathsf{clip}\left(\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i)\right) + \xi_t. \tag{6}$$

This means that the last iterate $\mathbf{m}_T$ is effectively the result of only a single iteration of DP-SGD. In particular, it will have a privacy guarantee corresponding to one iteration. Combining this regularizer with a linear loss and a setting of the parameters T, q, σ, δ such that the privacy loss is non-monotone – i.e., $\varepsilon_{T,q,\sigma}(\delta) < \varepsilon_{1,q,\sigma}(\delta)$ – yields a counterexample.

In light of this counterexample, in the next subsection, we benchmark our counterexample against sweeping over smaller values of T. I.e., we consider $\max_{t \le T} \varepsilon_{t,q,\sigma}(\delta)$ instead of simply $\varepsilon_{T,q,\sigma}(\delta)$.

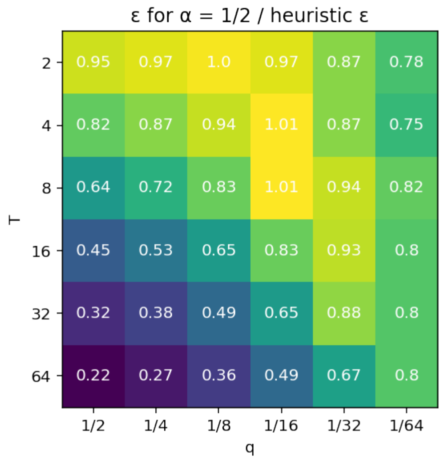
4.2 Linear Loss + Quadratic Regularizer

Consider running DP-SGD in one dimension (i.e., $d = 1$) with a linear loss $\ell(\mathbf{m}, x) = \mathbf{m}x$ for the canary and a quadratic regularizer $r(\mathbf{m}) = \frac{1}{2}\alpha\mathbf{m}^2$, where $\alpha \in [0, 1]$ and $x \in [-1, 1]$ and we use learning rate $\eta = 1$. With sampling probability q, after T iterations the privacy guarantee is equivalent to distinguishing $Q := \mathcal{N}(0, \widehat{\sigma}^2)$ and $P := \mathcal{N}(\sum_{i \in [T]} (1-\alpha)^{i-1} \mathsf{Bernoulli}(q), \widehat{\sigma}^2)$, where $\widehat{\sigma}^2 := \sigma^2 \sum_{i \in [T]} (1-\alpha)^{2(i-1)}$. When $\alpha = 0$, this retrieves linear losses. When $\alpha = 1$, this corresponds to distinguishing $\mathcal{N}(0, \widehat{\sigma}^2)$ and $\mathcal{N}(\mathsf{Bernoulli}(q), \widehat{\sigma}^2)$ or, equivalently, to distinguishing linear losses after $T = 1$ iteration. If we maximize our heuristic over the number of iterations $\le T$, then our heuristic is tight for the extremes $\alpha \in \{0, 1\}$.

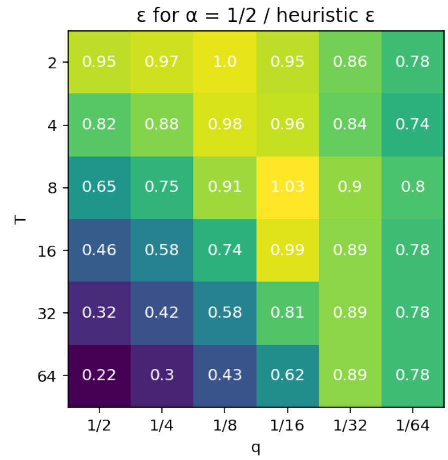
A natural question is whether the worst-case privacy guarantee on this quadratic is always given by $\alpha \in \{0, 1\}$. Perhaps surprisingly, the answer is no: we found that for $T = 3, q = 0.1, \sigma = 1, \alpha = 0$, DP-SGD is $(2.222, 10^{-6})$-DP. For $\alpha = 1$ instead DP-SGD is $(2.182, 10^{-6})$-DP. However, for $\alpha = 0.5$ instead the quadratic loss does not satisfy $(\varepsilon, 10^{-6})$-DP for $\varepsilon < 2.274$.

However, this violation is small, which suggests our heuristic is still a reasonable for this class of examples. To validate this, we consider a set of values for the tuple (T, q, σ). For each setting of T, q, σ, we compute $\max_{t \le T} \varepsilon_{t,q,\sigma}(\delta)$ at $\delta = 10^{-6}$. We then compute ε for the linear loss with quadratic regularizer example with $\alpha = 1/2$ in the same setting. Since the support of the random variable $\sum_{i \in [T]} (1-\alpha)^{i-1} \mathsf{Bernoulli}(q)$ has size 2^T for $\alpha = 1/2$, computing exact ε for even moderate T is computationally intensive. Instead, let X be the random variable equal to $\sum_{i \in [T]} (1-\alpha)^{i-1} \mathsf{Bernoulli}(q)$, except we round up values in the support which are less than .0005 up to .0005, and then round each value in the support up to the nearest integer power of 1.05. We then compute an exact ε for distingushing $\mathcal{N}(0, \widehat{\sigma}^2)$ vs $\mathcal{N}(X, \widehat{\sigma}^2)$. By Lemma 4.5 of Choquette-Choo, Ganesh, Steinke, and Thakurta [CCGST24], we know that distinguishing $\mathcal{N}(0, \widehat{\sigma}^2)$ vs. $\mathcal{N}(\sum_{i \in [T]} (1-\alpha)^{i-1} \mathsf{Bernoulli}(q), \widehat{\sigma}^2)$ is no harder than distingushing $\mathcal{N}(0, \widehat{\sigma}^2)$ vs $\mathcal{N}(X, \widehat{\sigma}^2)$, and since we increase the values in the support by no more than 1.05 multiplicatively, we expect that our rounding does not increase ε by more than 1.05 multiplicatively.

In Figure 5, we plot the ratio of ε at $\delta = 10^{-6}$ for distingushing between $\mathcal{N}(0, \widehat{\sigma}^2)$ and $\mathcal{N}(X, \widehat{\sigma}^2)$ divided by the maximum over $i \in [T]$ of ε at $\delta = 10^{-6}$ for distinguishing between $\mathcal{N}(0, i\sigma^2)$ and $\mathcal{N}(\mathsf{Binomial}(i, q), i\sigma^2)$. We sweep over T and q, and for each q In Figure 5a (resp. Figure 5b) we set σ such that distinguishing $\mathcal{N}(0, \sigma^2)$ from $\mathcal{N}(\mathsf{Bernoulli}(q), \sigma^2)$ satisfies $(1, 10^{-6})$-DP



(a) One iteration of DP-SGD satisfies $(1, 10^{-6})$-DP.



(b) One iteration of DP-SGD satisfies $(2, 10^{-6})$-DP.

Figure 5: Ratio of upper bound on ε for quadratic loss with $\alpha = 0.5$ divided by maximum ε of i iterations on a linear loss. In Figure 5a (resp. Figure 5b), for each choice of q, σ is set so 1 iteration of DP-SGD satisfies $(1, 10^{-6})$-DP (resp $(2, 10^{-6})$-DP).

(resp. $(2, 10^{-6})$-DP). In the majority of settings, the linear loss heuristic provides a larger ε than the quadratic with $\alpha = 1/2$, and even when the quadratic provides a larger ε, the violation is small ($\leq 3\%$). This is evidence that our heuristic is still a good approximation for many convex losses.

4.3 Pathological Example

If we allow the regularizer r to be arbitrary – in particular, not even requiring continuity – then the gradient can also be arbitrary. This flexibility allows us to construct a counterexample such that the standard composition-based analysis of DP-SGD with `intermediate_iterates` is close to tight.

Specifically, choose the regularizer so that the update $\mathbf{m}' = \mathbf{m} - \eta \nabla_{\mathbf{m}} r(\mathbf{m})$ does the following: $\mathbf{m}'_1 = 0$ and, for $i \in [d-1]$, $\mathbf{m}'_{i+1} = v \cdot \mathbf{m}_i$. Here $v > 1$ is a large constant. We chose the loss so that, for our canary x_1, we have $\nabla_{\mathbf{m}} \ell(\mathbf{m}, x_1) = (1, 0, 0, \cdots, 0)$ and, for all other examples x_i ($i \in \{2, 3, \cdots, n\}$), we have $\nabla_{\mathbf{m}} \ell(\mathbf{m}, x_i) = \mathbf{0}$. Then the last iterate is

$$\mathbf{m}_T = (A_T + \xi_{T,1}, vA_{T-1} + v\xi_{T-1,1} + \xi_{T,2}, v^2 A_{T-2} + v^2 \xi_{T-2,1} + v\xi_{T-1,2} + \xi_{T,3}, \cdots), \quad (7)$$

where $A_t \leftarrow \mathsf{Bernoulli}(p)$ indicates whether or not the canary was sampled in the t-th iteration and $\xi_{t,i}$ denotes the i-th coordinate of the noise ξ_t added in the t-th step. Essentially, the last iterate $\mathbf{m}_T$ contains the history of all the iterates in its coordinates. Namely, the i-th coordinate of $\mathbf{m}_T$ gives a scaled noisy approximation to A_{T-i}:

$$v^{1-i} \mathbf{m}_{T,i} = A_{T-i} + \sum_{j=0}^{i-1} v^{j+1-i} \xi_{T-j,i-j} \sim \mathcal{N}\left(A_{T-i}, \sigma^2 \frac{1 - v^{-2i}}{1 - v^{-2}}\right). \quad (8)$$

As $v \to \infty$, the variance converges to σ^2. In other words, if v is large, from the final iterate, we can obtain $\mathcal{N}(A_i, \sigma^2)$ for all i. This makes the standard composition-based analysis of DP-SGD tight.

4.4 Malicious Dataset Attack

The examples above rely on the regularizer having large unclipped gradients. We now construct a counterexample without a regularizer, instead using other examples to amplify the canary signal.

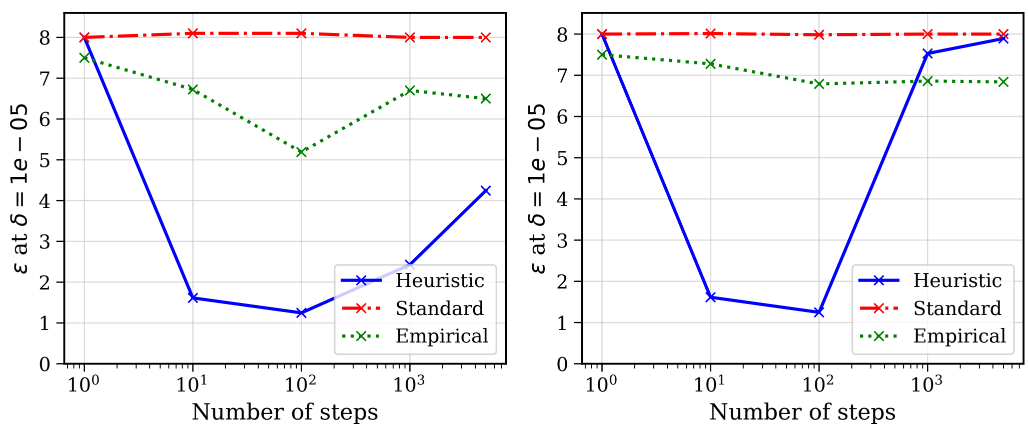
Our heuristic assumes the adversary does not have access to the intermediate iterations and that the model is linear. However, we can design a nonlinear model and specific training data to directly challenge this assumption. The attack strategy is to use the model's parameters as a sort of noisy storage, saving all iterations within them. Then with access only to the final model, an adversary

Table 1: Previous works showed that large batch sizes achieve high performing models [DBHSB22]. Using our heuristic analysis it is possible to achieve similar performance for smaller batch sizes.

Batch size	Heuristic ε	Standard ε	Accuracy	Empirical ε
4096	6.34	8	79.5%	1.7
512	7.0	12	79.1%	1.8
256	6.7	14	79.4%	1.6

can still examine the parameters, extract the intermediate steps, and break the assumption. Our construction introduces a data point that changes its gradient based on the number of past iterations, making it easy to identify if the point was present a given iteration of training. The rest of the data points are maliciously selected to ensure the noise added during training doesn't impact the information stored in the model's parameters. We defer the full details of the attack to Appendix C.

Figure 6 summarizes the results. As illustrated in the figure, this attack achieves a auditing lower bound matching the standard DP-SGD analysis even in the `last_iterate_only` setting. As a result, the attack exceeds our heuristic. However, this is a highly artificial example and it is unlikely to reflect real-world scenarios.



(a) $q = 0.01$ (b) $q = 0.1$

Figure 6: In this adversarial example, the attack encodes all training steps within the final model parameters, thereby violating the specific assumptions used to justify our heuristic analysis.

5 Discussion & Conclusion

Both theoretical analysis and privacy auditing are valuable for understanding privacy leakage in machine learning, but each has limitations. Theoretical analysis is inherently conservative, while auditing procedures evaluate only specific attacks, and may thus underrepresent the privacy leakage.

Our work introduces a novel heuristic analysis for DP-SGD that focuses on the privacy implications of releasing only the final model iterate. This approach is based in the empirical observation that linear loss functions accurately model the effectiveness of state of the art membership inference attacks. Our heuristic offers a practical and computationally efficient way to estimate privacy leakage to complement privacy auditing and the standard composition-based analysis of DP-SGD. As shown in Table 1, we trained a series of CIFAR10 models with varying batch sizes that all achieved the similar level of heuristic epsilon, albeit with different standard epsilon values. Remarkably, these models exhibited similar performance and similar empirical epsilon values.

We also acknowledge the limitations of our heuristic by identifying specific counterexamples where the heuristic underestimates the true privacy leakage.

References

[ACGMMTZ16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318. URL: https://arxiv.org/abs/1607.00133 (cit. on pp. 1, 3).

[AKOOMS23] G. Andrew, P. Kairouz, S. Oh, A. Oprea, H. B. McMahan, and V. Suriyakumar. "One-shot Empirical Privacy Estimation for Federated Learning". In: *arXiv preprint arXiv:2302.03098* (2023). URL: https://arxiv.org/abs/2302.03098 (cit. on p. 2).

[Ale09] K. Alex. "Learning multiple layers of features from tiny images". In: *https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf* (2009) (cit. on pp. 5, 20).

[AT22] J. Altschuler and K. Talwar. "Privacy of noisy stochastic gradient descent: More iterations without more privacy loss". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 3788–3800. URL: https://arxiv.org/abs/2205.13710 (cit. on p. 2).

[AZT24] M. Aerni, J. Zhang, and F. Tramèr. "Evaluations of Machine Learning Privacy Defenses are Misleading". In: *arXiv preprint arXiv:2404.17399* (2024). URL: https://arxiv.org/abs/2404.17399 (cit. on p. 2).

[BGDCTV18] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. Vechev. "Dp-finder: Finding differential privacy violations by sampling and optimization". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018, pp. 508–524 (cit. on p. 2).

[BSA24] J. Bok, W. Su, and J. M. Altschuler. "Shifted Interpolation for Differential Privacy". In: *arXiv preprint arXiv:2403.00278* (2024). URL: https://arxiv.org/abs/2403.00278 (cit. on p. 2).

[BST14] R. Bassily, A. Smith, and A. Thakurta. "Private empirical risk minimization: Efficient algorithms and tight error bounds". In: *2014 IEEE 55th annual symposium on foundations of computer science*. IEEE. 2014, pp. 464–473. URL: https://arxiv.org/abs/1405.7085 (cit. on pp. 1, 3).

[BTRKMW24] M. Bertran, S. Tang, A. Roth, M. Kearns, J. H. Morgenstern, and S. Z. Wu. "Scalable membership inference attacks via quantile regression". In: *Advances in Neural Information Processing Systems* 36 (2024). URL: https://arxiv.org/abs/2307.03694 (cit. on p. 2).

[CCGST24] C. A. Choquette-Choo, A. Ganesh, T. Steinke, and A. Thakurta. *Privacy Amplification for Matrix Mechanisms*. 2024. arXiv: 2310.15526 [cs.LG] (cit. on p. 7).

[CCNSTT22] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. "Membership inference attacks from first principles". In: *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2022, pp. 1897–1914. URL: https://arxiv.org/abs/2112.03570 (cit. on p. 2).

[CYS21] R. Chourasia, J. Ye, and R. Shokri. "Differential privacy dynamics of langevin diffusion and noisy gradient descent". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14771–14781. URL: https://arxiv.org/abs/2102.05855 (cit. on p. 2).

[DBHSB22] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle. "Unlocking high-accuracy differentially private image classification through scale". In: *arXiv preprint arXiv:2204.13650* (2022) (cit. on pp. 5, 9).

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "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*. Springer. 2006, pp. 265–284. URL: https://www.iacr.org/archive/tcc2006/38760266/38760266.pdf (cit. on p. 1).

[DRS19] J. Dong, A. Roth, and W. J. Su. "Gaussian differential privacy". In: *arXiv preprint arXiv:1905.02383* (2019) (cit. on p. 13).

[DSSUV15] C. Dwork, A. Smith, T. Steinke, J. Ullman, and S. Vadhan. "Robust traceability from trace amounts". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 650–669 (cit. on p. 2).

[DWWZK18] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer. "Detecting violations of differential privacy". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018, pp. 475–489 (cit. on p. 2).

[FMTT18] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta. "Privacy amplification by iteration". In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2018, pp. 521–532. URL: https://arxiv.org/abs/1808.06651 (cit. on p. 2).

[Goo20] Google. *Differential Privacy Accounting*. https://github.com/google/differential-privacy/tree/main/python/dp_accounting. 2020 (cit. on pp. 3, 4).

[HSRDTMPSNC08] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". In: *PLoS genetics* 4.8 (2008), e1000167 (cit. on p. 2).

[JUO20] M. Jagielski, J. Ullman, and A. Oprea. "Auditing differentially private machine learning: How private is private sgd?" In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22205–22216 (cit. on p. 2).

[KJH20] A. Koskela, J. Jälkö, and A. Honkela. "Computing tight differential privacy guarantees using fft". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2560–2569. URL: https://arxiv.org/abs/1906.03049 (cit. on p. 1).

[LF20] K. Leino and M. Fredrikson. "Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference". In: *29th USENIX security symposium (USENIX Security 20)*. 2020, pp. 1605–1622. URL: https://arxiv.org/abs/1906.11798 (cit. on p. 2).

[Mir17] I. Mironov. "Rényi differential privacy". In: *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE. 2017, pp. 263–275. URL: https://arxiv.org/abs/1702.07476 (cit. on p. 1).

[NHSBTJCT23] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. "Tight Auditing of Differentially Private Machine Learning". In: *arXiv preprint arXiv:2302.07956* (2023). URL: https://arxiv.org/abs/2302.07956 (cit. on pp. 1, 2, 5, 6).

[NSTPC21] M. Nasr, S. Song, A. Thakurta, N. Papernot, and N. Carlini. "Adversary instantiation: Lower bounds for differentially private machine learning". In: *2021 IEEE Symposium on security and privacy (SP)*. IEEE. 2021, pp. 866–882. URL: https://arxiv.org/abs/2101.04535 (cit. on pp. 1, 2, 5, 6).

[SDSOJ19] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou. "White-box vs black-box: Bayes optimal strategies for membership inference". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5558–5567. URL: https://arxiv.org/abs/1908.11229 (cit. on p. 2).

[SNJ23] T. Steinke, M. Nasr, and M. Jagielski. "Privacy auditing with one (1) training run". In: *Advances in Neural Information Processing Systems* 36 (2023). URL: https://arxiv.org/abs/2305.08846 (cit. on pp. 2, 6).

[SOJH09] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin. "Genomic privacy and limits of individual detection in a pool". In: *Nature genetics* 41.9 (2009), pp. 965–967 (cit. on p. 2).

[SSSS17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership inference attacks against machine learning models". In: *2017 IEEE symposium on security and privacy (SP)*. IEEE. 2017, pp. 3–18 (cit. on p. 2).

[Ste22] T. Steinke. "Composition of Differential Privacy & Privacy Amplification by Subsampling". In: *arXiv preprint arXiv:2210.00597* (2022). URL: https://arxiv.org/abs/2210.00597 (cit. on p. 1).

[TTSSJC22] F. Tramer, A. Terzis, T. Steinke, S. Song, M. Jagielski, and N. Carlini. "Debugging differential privacy: A case study for privacy auditing". In: *arXiv preprint arXiv:2202.12219* (2022). URL: https://arxiv.org/abs/2202.12219 (cit. on p. 2).

[WBKBGGG23] Y. Wen, A. Bansal, H. Kazemi, E. Borgnia, M. Goldblum, J. Geiping, and T. Goldstein. "Canary in a coalmine: Better membership inference with ensembled adversarial queries". In: *ICLR*. 2023. URL: https://arxiv.org/abs/2210.10750 (cit. on p. 2).

[YS22] J. Ye and R. Shokri. "Differentially private learning needs hidden state (or much faster convergence)". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 703–715. URL: https://arxiv.org/abs/2203.05363 (cit. on p. 2).

[ZBWTSRPNK22] S. Zanella-Béguelin, L. Wutschitz, S. Tople, A. Salem, V. Rühle, A. Paverd, M. Naseri, and B. Köpf. "Bayesian estimation of differential privacy". In: *arXiv preprint arXiv:2206.05199* (2022) (cit. on p. 2).

[ZK16] S. Zagoruyko and N. Komodakis. "Wide residual networks". In: *arXiv preprint arXiv:1605.07146* (2016) (cit. on pp. 5, 20).

[ZLS23] S. Zarifzadeh, P. C.-J. M. Liu, and R. Shokri. "Low-Cost High-Power Membership Inference by Boosting Relativity". In: (2023). URL: https://arxiv.org/abs/2312.03262 (cit. on p. 2).

A Proof of Theorem 1

Proof. Let x_{i^*} be the canary, let D be the dataset with the canary and D' be the dataset without the canary. Since ℓ and r are linear, wlog we can assume $r = 0$ and $\nabla_{\mathbf{m}_{t-1}} \ell(\mathbf{m}_{t-1}, x_i) = \mathbf{v}_i$ for some set of vectors $\{\mathbf{v}_i\}$, such that $\|\mathbf{v}_i\|_2 \leq 1$. We can also assume wlog $\|\mathbf{v}_{i^*}\| = 1$ since, if $\|\mathbf{v}_{i^*}\| < 1$, the final privacy guarantee we show only improves.

We have the following recursion for $\mathbf{m}_t$:

$$\mathbf{m}_t = \mathbf{m}_{t-1} - \eta \left(\sum_{i \in B_t} \mathbf{v}_i + \xi_t \right), \qquad \xi_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2 I_d).$$

Unrolling the recursion:

$$\mathbf{m}_t = \mathbf{m}_0 - \eta \left[\sum_{t \in [T]} \sum_{i \in B_t} \mathbf{v}_i + \xi \right], \qquad \xi \sim N(0, T\sigma^2 I_d).$$

By the post-processing property of DP, we can assume that in addition to the final model $\mathbf{m}_T$, we release $\mathbf{m}_0$ and $\{B_t \setminus \{x_{i^*}\}\}_{t \in [T]}$, that is we release all examples that were sampled in each batch except for the canary. The following f is a bijection, computable by an adversary using the released information:

$$f(\mathbf{m}_T) := -\left[\frac{\mathbf{m}_T - \mathbf{m}_0}{\eta} - \sum_{t \in [T]} \sum_{i \in B_t \setminus \{x_{i^*}\}} \mathbf{v}_i. \right]$$

Since f is a bijection, distinguishing $\mathbf{m}_T$ sampled using D and D' is equivalent to distinguishing $f(\mathbf{m}_T)$ instead. Now we have $f(\mathbf{m}_T) = \mathcal{N}(0, T\sigma^2 I_d)$ for D', and $f(\mathbf{m}_T) = \mathcal{N}(0, T\sigma^2 I_d) + k\mathbf{v}_{i^*}$, $k \sim \mathsf{Binomial}(T, q)$. For any vector $\mathbf{u}$ orthogonal to $\mathbf{v}_{i^*}$, by isotropy of the Gaussian distribution the distribution of $\langle f(\mathbf{m}_T), \mathbf{u} \rangle$ is the same for both D and D' and independent of $\langle f(\mathbf{m}_T), \mathbf{v}_{i^*} \rangle$, hence distinguishing $f(\mathbf{m}_T)$ given D and D' is the same as distingushing $\langle f(\mathbf{m}_T), \mathbf{v}_{i^*} \rangle$ given D and D'. Finally, the distribution of $\langle f(\mathbf{m}_T), \mathbf{v}_{i^*} \rangle$ is exactly P for D and exactly Q for D'. By post-processing, this gives the theorem.

We can also see that the function $\delta_{T,q,\sigma}$ is tight (i.e., even if we do not release $B_t \setminus \{x_{i^*}\}$), by considering the 1-dimensional setting, where $\mathbf{v}_i = 0$ for $i \neq i^*$ and $\mathbf{v}_{i^*} = -1, \eta = 1, \mathbf{m}_0 = 0$. Then, the distribution of $\mathbf{m}_T$ given D is exactly P, and given D' is exactly Q. $\square$

A.1 Computing δ from ε

Here, we give an efficiently computable expression for the function $\delta_{T,q,\sigma}(\varepsilon)$. Using P, Q as in Theorem 1, let $f(y)$ be the privacy loss for the output y:

$$\begin{aligned} f(y) = \log\left(\frac{P(y)}{Q(y)}\right) &= \log\left(\sum_{k=0}^{T}\binom{T}{k}q^k(1-q)^{n-k}\frac{\exp(-(y-k)^2/2T\sigma^2)}{\exp(-y^2/2T\sigma^2)}\right) \\ &= \log\left(\sum_{k=0}^{T}\binom{T}{k}q^k(1-q)^k\exp\left(\frac{2ky-k^2}{2T\sigma^2}\right)\right). \end{aligned}$$

Then for any ε, using the fact that $S = \{y : f(y) \geq \varepsilon\}$ maximizes $P(S) - e^\varepsilon Q(S)$, we have:

$$\begin{aligned} H_{e^\varepsilon}(P,Q) &= P(\{y : f(y) \geq \varepsilon\}) - e^\varepsilon Q(\{y : f(y) \geq \varepsilon\}) \\ &= P(\{y : y \geq f^{-1}(\varepsilon)\}) - e^\varepsilon Q(\{y : y \geq f^{-1}(\varepsilon)\}) \\ &= \sum_{k=0}^{T}\binom{T}{k}q^k(1-q)^k \Pr[\mathcal{N}(k, T\sigma^2) \geq f^{-1}(\varepsilon)] - e^\varepsilon \Pr[\mathcal{N}(0, T\sigma^2) \geq f^{-1}(\varepsilon)]. \end{aligned}$$

Similarly, $S = \{y : f(y) \leq -\varepsilon\}$ maximizes $Q(S) - e^\varepsilon P(S)$ so we have:

$$\begin{aligned} H_{e^\varepsilon}(Q,P) &= Q(\{y : f(y) \leq -\varepsilon\}) - e^\varepsilon P(\{y : f(y) \leq -\varepsilon\}) \\ &= Q(\{y : y \leq f^{-1}(-\varepsilon)\}) - e^\varepsilon P(\{y : y \leq f^{-1}(-\varepsilon)\}) \\ &= \Pr[\mathcal{N}(0, T\sigma^2) \leq f^{-1}(-\varepsilon)] - e^\varepsilon \sum_{k=0}^{T}\binom{T}{k}q^k(1-q)^k \Pr[\mathcal{N}(k, T\sigma^2) \leq f^{-1}(-\varepsilon)]. \end{aligned}$$

These expressions can be evaluated efficiently. Since f is monotone, it can be inverted via binary search. We can also use binary search to evaluate ε as a function of δ.

B Linear Worst Case for Full Batch Setting

It turns out that in the full-batch setting, the worst-case analyses of DP-GD with `intermediate_iterates` and with `last_iterate_only` are the same. This phenomenon arises because there is no subsampling (because $q = 1$ in Algorithm 1) and thus the algorithm is "just" the Gaussian mechanism. Intuitively, DP-GD with `intermediate_iterates` corresponds to T calls to the Gaussian mechanism with noise multiplier σ, while DP-GD with `last_iterate_only` corresponds to one call to the Gaussian mechanism with noise multiplier $\sigma/\sqrt{T}$; these are equivalent by the properties of the Gaussian distribution.

We can formalize this using the language of Gaussian DP [DRS19]: DP-GD (Algorithm 1 with $q = 1$) satisfies $\sqrt{T}/\sigma$-GDP. (Each iteration satisfies $1/\sigma$-GDP and adaptive composition implies the overall guarantee.) This means that the privacy loss is exactly dominated by that of the Gaussian mechanism with noise multiplier $\sigma/\sqrt{T}$. Linear losses give an example such that DP-GD with `last_iterate_only` has exactly this privacy loss, since the final iterate reveals the sum of all the noisy gradient estimates. The worst-case privacy of DP-GD with `intermediate_iterates` is no worse than that of DP-GD with `last_iterate_only`. The reverse is also true (by postprocessing).

In more detail: For T iterations of (full-batch) DP-GD on a linear losses, if the losses are (wlog) 1-Lipschitz and we add noise $\mathcal{N}(0, \frac{\sigma^2}{n^2} \cdot I)$ to the gradient in every round, distinguishing the last iterate of DP-SGD on adjacent databases is equivalent to distinguishing $\mathcal{N}(0, T\sigma^2)$ and $\mathcal{N}(T, T\sigma^2)$. This can be seen as a special case of Theorem 1 for $p = 1$, so we do not a give a detailed argument here.

If instead we are given every iteration $\mathbf{m}_t$, for *any* 1-Lipschitz loss, distinguishing the joint distributions of $\mathbf{m}_t$ given $\mathbf{m}_{t-1}$ on adjacent databases is equivalent to distinguishing $\mathcal{N}(0, \sigma^2)$ and $\mathcal{N}(1, \sigma^2)$. In turn, distinguishing the distribution of all iterates on adjacent databases is equivalent to distinguishing $\mathcal{N}(\mathbf{0}^T, \sigma^2 I_T)$ and $\mathcal{N}(\mathbf{1}^T, \sigma^2 I_T)$, where $\mathbf{0}^T$ and $\mathbf{1}^T$ are the all-zeros and all-ones vectors in $\mathbb{R}^T$. Because the Gaussian distribution is isotropic, distinguishing $\mathcal{N}(\mathbf{0}^T, \sigma^2 I_T)$ and $\mathcal{N}(\mathbf{1}^T, \sigma^2 I_T)$ is equivalent to distinguishing $\langle \mathbf{x}, \mathbf{1}^T \rangle$ where $\mathbf{x} \sim \mathcal{N}(\mathbf{0}^T, \sigma^2 I_T)$ and $\langle \mathbf{x}, \mathbf{1}^T \rangle$ where $\mathbf{x} \sim \mathcal{N}(\mathbf{1}^T, \sigma^2 I_T)$. These distributions are $\mathcal{N}(0, T\sigma^2)$ and $\mathcal{N}(T, T\sigma^2)$, the exact pair of distributions we reduced to for last-iterate analysis of linear losses.

C Malicious Dataset Attack Details

Algorithms 2, 3, and 4 summarizes the construction for the attack. The attack assume the model parameters have dimension equal to the number of iterations. It also assumes each data point can reference which iteration of training is currently happening (this can be implemented by having a single model parameter which increments in each step, independently of the training examples, without impacting the privacy of the training process). Then we build our two datasets D and $D' = D \cup \{x\}$ so that all points in dataset D ("repeaters") run Algorithm 3 to compute gradients and the canary point in D' runs Algorithm 2 to compute its gradient. Our attack relies heavily on DP-SGD's lack of assumptions on the data distribution and any specific properties of the model or gradients. Algorithm 2, which generates the canary data point, is straightforward. Its goal is to store in the model parameters whether it was present in iteration i by outputting a gradient that changes only the i-th index of the model parameters by 1 (assuming a clipping threshold of 1).

All other data points, the "repeaters", are present in both datasets (D and D'), and have three tasks:

- Cancel out any noise added to the model parameters at an index larger than the current iteration. At iteration i, their gradients for parameters from index i onward will be the same as the current value of the model parameter, scaled by the batch size and the learning rate to ensure this parameter value will be 0 after the update.
- Evaluate whether the canary point was present in the previous iteration by comparing the model parameter at index $i - 1$ with a threshold, and rewrite the value of that model parameter to a large value if the canary was present.
- Ensure that all previous decisions are not overwritten by noise by continuing to rewrite them with a large value based on their previous value.

To achieve all of these goals simultaneously, we require that the batch size is large enough that the repeaters' updates are not clipped.

Finally Algorithm 4 runs DP-SGD, with repeater points computing gradients with Algorithm 3 and the canary point, sampled with probability p, computing its gradient using Algorithm 2. In our experiments we run Algorithm 4 100,000 times. And to evaluate if the model parameters was from dataset D or D' we run a hypothesis test on the values of the model parameters. All constants are chosen to ensure all objectives of the repeaters are satisfied.

Algorithm 2 Canary data point

```
1: function ADV(x, i)
2:     Initialize a as a zero vector of the same dimension as x
3:     Set a_i ← 1                                  ▷ Set the i-th component to 1
4:     return −a
5: end function
```

Algorithm 3 Additional data points

Require: model parameters $\mathbf{x}$, iteration number i, batch size N, learning rate η, previous history threshold t_{past}, last iteration threshold t_{last}, history amplification value BIG_VAL

function REPEATERS($\mathbf{x}$, i, N, η, t_{past}, t_{last}, BIG_VAL)
 $\mathbf{h} \leftarrow \mathbf{x}_{0:i}$ ▷ Parameter "history" up to iteration i, not inclusive
 $\mathbf{f} \leftarrow \mathbf{x}_{i:end}$ ▷ Future and current parameters, starting from iteration i
 $\mathbf{f} \leftarrow -\mathbf{f}/(\eta \cdot N)$ ▷ Remove noise from last iteration
 base_history $\leftarrow -\mathbf{x}_{0:i}/(\eta \cdot N)$ ▷ By default, zero out entire history
 if length$(\mathbf{h}) > 1$ **then**
  $\mathbf{h}_{0:i-1} \leftarrow$ BIG_VAL$/(\eta \cdot N) \cdot (2\mathbb{1}[\mathbf{h}_{0:i-1} \geq t_{\text{past}}] - 1)$ ▷ If an old iteration is large enough, it was a canary iteration, so amplify it
 end if
 if length$(\mathbf{h}) > 0$ **then**
  $\mathbf{h}_{i-1} \leftarrow$ BIG_VAL$/(\eta \cdot N) \cdot (2\mathbb{1}[\mathbf{h}_i \geq t_{\text{last}}] - 1)$ ▷ If the last iteration is large enough, it was a canary iteration, so amplify it
 end if
 $\mathbf{h} \leftarrow \mathbf{h} +$ base_history ▷ Don't zero out canary iterations
 $\mathbf{a} \leftarrow$ concatenate$(\mathbf{h}, \mathbf{f})$
 return $-\mathbf{a}$
end function

Algorithm 4 Encoding Attacking

Require: add-diff, whether to add the canary, batch size N, sampling rate p, learning rate (η), iteration count/parameter count D

function RUN_DPSGD(add-diff)
 $C \leftarrow 1$
 Initialize model $\mathbf{m} \leftarrow \mathbf{0}$ of dimension D
 for $i = 0$ to D **do**
  Generate a uniform random value $q \in [0, 1]$
  $\mathbf{r} \leftarrow$ repeaters$(\mathbf{m}, i)$
  Compute norm $c \leftarrow ||\mathbf{r}||$
  if $c > 0$ **then**
   Normalize $\mathbf{r} \leftarrow \mathbf{r}/\max(c, C)$
  end if
  Adjusted vector $\mathbf{z} \leftarrow \mathbf{r} \times N$
  Verify condition on $\mathbf{m}_i$
  if $p \leq q$ and add-diff **then**
   $\mathbf{r} \leftarrow$ adv$(\mathbf{m}, i)$
   Normalize and update $\mathbf{z}$
  end if
  Apply Gaussian noise to $\mathbf{z}$
  Update model $\mathbf{m} \leftarrow \mathbf{m} - \mathbf{z} \times \eta$
 end for
 return $\mathbf{m}$
end function

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state what we do and then the following sections and the appendix provide details.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4 discusses the limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theorem 1 is proved in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The setup is described for each experiment we conduct and we reference prior work that these build on.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We intend to release the code eventually, but we are not able to do so at the moment; we refrain from providing a detailed reason, as this could violate anonymity.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The setup is described for each experiment we conduct and we reference prior work that these build on. We use the standard CIFAR10 dataset for deep learning experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The auditing results present a lower bound which can be viewed as a one-sided confidence interval. For the other results the numbers are computed non-statistically (i.e. by numerically evaluating a formula); the only potential error here is due to numerical precision.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We used A2-megagpu-16g machines from Google cloud which have 16 Nvidia A100 40GB GPUs to run the experiments in this paper. Overall we used around 33,000 hours of GPU to run all of the experiments in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: No human subjects or sensitive data were used.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is primarily theoretical. While it is possible that downstream uses of our work could be societally impactful, the precise consequences are difficult to foresee. The considerations are similar to any other paper on private machine learning.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper uses standard datasets (CIFAR10) and standard models (WideResNet).

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use CIFAR10 [Ale09] and a WideResNet [ZK16].

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our main contribution is a heuristic privacy analysis. This is fully described in the paper and can be computed using existing open-source libraries.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.