Revisiting the Provable-Auditable Privacy Gap of DP-SGD

Saloni Modi
University of Texas at Austin
saloni.a.modi@utexas.edu

Srivi Balaji*
University of Texas at Austin
srivibalaji@utexas.edu

Yusong Zhu*
University of Texas at Austin
zhuys@utexas.edu

Gautam Kamath
University of Waterloo and Vector Institute
g@csail.mit.edu

Kevin Tian
University of Texas at Austin
kjtian@cs.utexas.edu

Abstract

Differential privacy (DP) has traditionally been used to provide theoretical upper bounds on an algorithm's stability to changing its training data. In modern machine learning applications, achieving strong tradeoffs between utility and theoretical privacy is challenging, and thus one may optimistically hope that existing theoretical analyses are loose. Recent work on *privacy auditing* has adopted a dual viewpoint, instead lower bounding the true privacy of an algorithm by constructing empirical distinguishing events. The auditing literature has thus far yielded a pessimistic outlook on the looseness of theoretical privacy bounds for DP-SGD, the de facto private training method in modern ML, as nearly-matching empirical lower bounds have been achieved under various threat models [NHS+23, AC24, CBP25].

In this work, we propose the empirical privacy lower bound of an algorithm as a concrete metric to optimize for, complementary to the theoretical upper bound. We give a lightweight defense framework that generically augments optimization methods in the ML pipeline to have significantly-improved empirical privacy on standard benchmarks. Moreover, we show that our framework comes at *no theoretical privacy cost* when augmenting DP-SGD, unlike previous defenses against membership inference attacks. We evaluate our defense against a broad range of audit constructions, models, and datasets to demonstrate its flexibility. Our code implementation can be found at this repository.

1 Introduction

It is well-documented that machine learning (ML) models are vulnerable to a variety of privacy threats [HSR+08, SSSS17, CTW+21]. To combat these threats, differential privacy (DP) [DMNS06] has seen wide adoption as a rigorous notion of data privacy. This notion is typically phrased as an upper bound (Definition 1): for an algorithm $\mathcal{A} : \mathcal{S}^* \to \Omega$, DP bounds the value of

$$\varepsilon^\star(\delta) := \log \left(\sup_{\substack{\mathcal{D},\mathcal{D}' \in \mathcal{S}^* \\ \text{neighboring}}} \sup_{\mathcal{E} \subseteq \Omega} \frac{\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{E}] - \delta}{\Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{E}]} \right). \tag{1}$$

The quantity $\varepsilon^\star(\delta)$ can be viewed as the "true privacy" parameter at failure probability δ of $\mathcal{A}$, as it is realized by (or is the limit of) concrete neighboring $\mathcal{D}, \mathcal{D}'$,[2] and an "audit" $\mathcal{E} \in \Omega$. Various techniques

*Equal contribution.

[2]Different works in the privacy auditing literature use slightly different definitions of neighboring. In this work, we use the "add/remove" definition as in [MTZ19, NHS+23, CBP25]; see Section 2 for more discussion.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

have been developed to upper bound $\varepsilon^\star$. For example, modern DP-SGD [SCS13, BST14, ACG+16] privacy analyses proceed using (advanced) composition [DRV10, DR14], sometimes on an alternative metric such as Rényi differential privacy (RDP) [ACG+16, Mir17]. We informally use $\varepsilon_{\text{ub}}(\delta)$ to denote the tightest upper bound on (1) yielded by existing accounting techniques.

There are various ways that privacy accounting can be loose. A simple example is lossiness in converting between privacy definitions. A more subtle example, concerning the *threat model*, asks: what should we view as the output Ω of an algorithm, such as DP-SGD, that (adaptively) generates a sequence of private models $M_1, M_2, \ldots, M_T$? One way to define Ω is as a product over T model realizations. This is implicit in standard (composition-based) DP analyses, for which upper bounds on $\varepsilon^\star(\delta)$ hold even if $\mathcal{E}$ depends on all intermediate models. However, it is often more realistic to assume that only the final M_T is published (a.k.a. the "hidden state model" [YS22, CBP25]), or even that our access to M_T is limited (e.g., query access rather than full model weights), in which case the feasible audits $\mathcal{E}$ in (1) may be severely restricted. Indeed, recent theory has illustrated scenarios where changing the threat model can (significantly) amplify privacy [YS22, AT22].

1.1 Motivation

There is good reason to hope that current DP accounting significantly overestimates the true privacy $\varepsilon^\star$. It is well-documented that, to retain acceptable utility, current DP optimizers must compromise by yielding fairly large privacy parameters. For example, Figure 1(a) of [DBH+22] summarizes recent state-of-the-art accuracy of private training on CIFAR-10. With a strong provable privacy $\varepsilon_{\text{ub}} \approx 1$, the best accuracy was only 60% [TB21], compared with a non-private state of the art of 99%+. On the other hand achieving a modest accuracy of $> 80\%$ required a much larger $\varepsilon_{\text{ub}} \approx 8$. The worst-case implications of such a large ε_{ub} are dubious, as $\exp(8) \approx 3000$. Optimistically, could it be that the true privacy parameter is much smaller than current theory predicts?

A closely-related question is: how can a practitioner better reason about the true privacy $\varepsilon^\star$, taking the threat model into account? This is the key conceptual question that our work addresses. Motivated by work on privacy auditing [JUO20, NST+21], we propose using the following, more operational, "empirical privacy" quantity as an explicit metric for algorithm design:

$$\varepsilon_{\text{lb}}(\delta) := \log\left(\sup_{\mathcal{D}\in\mathcal{S}^n}\sup_{\text{canary } c}\sup_{\mathcal{E}\in\mathcal{T}}\max\left(\frac{\Pr[\mathcal{A}(\mathcal{D}^c)\in\mathcal{E}]-\delta}{\Pr[\mathcal{A}(\mathcal{D})\in\mathcal{E}]}, \frac{\Pr[\mathcal{A}(\mathcal{D})\in\mathcal{E}]-\delta}{\Pr[\mathcal{A}(\mathcal{D}^c)\in\mathcal{E}]}\right)\right). \tag{2}$$

In (2), $c \in \mathcal{S}$ is a "canary" sample, $\mathcal{T}$ denotes a family of audits, and the goal of $\mathcal{E}$ is to detect the presence of c in a dataset $\mathcal{D}$, where $\mathcal{D}^c := \mathcal{D} \cup \{c\}$. Under the strongest possible $\mathcal{D}$, c, and $\mathcal{E}$, the quantities (1) and (2) are the same. In practice, we cannot always find this optimal triple (e.g., the range Ω is typically infinite), and hence our estimated ε_{lb} is only a lower bound for $\varepsilon^\star$. Nonetheless, by optimizing over a comprehensive family of $(\mathcal{D}, c, \mathcal{E})$, one can hope to obtain nearly-tight *certifiable lower bounds* on $\varepsilon^\star(\delta)$. We review the standard auditing methodology in Appendix B.2. A typical setup involves a mislabeled or out-of-distribution canary c, and an audit that thresholds the loss value of the final model when labeling c. Intuitively, if c was memorized, then it will be correctly classified and hence will incur much lower loss than had the model never seen c.

Existing work on auditing DP-SGD, the de facto private training algorithm in ML, is surprisingly pessimistic: nearly-tight audits (e.g., certified ε_{lb} within a 30% factor of ε_{ub}) are achievable in a broad range of threat models.[3] This was first demonstrated by [NST+21, NHS+23] in the strong "gradient space canary" threat model, where the attacker can arbitrarily control *gradients* associated with a sample. Existing ε_{ub} bounds hold in this threat model, but potentially overestimate the true (1), in real-world settings where gradient space attacks are infeasible [CBP25, BLIW26]. Later, [AC24, CBP25] obtained similar nearly-tight audits under "hidden state" threat models where only the final model is released; particularly, [AC24] focuses on a more realistic "input space canary" threat model, where the canary is a fixed planted sample image.

1.2 Our contributions

Our work makes two main contributions towards more accurately measuring the true privacy (1), and designing algorithms with improved empirical privacy under realistic threat models.

[3] In Section 2, we formally define and review different threat models that we consider from the literature.

- We initiate the study of the *auditable privacy lower bound* ε_{lb} as a formal metric for algorithms to target. This line of research should be viewed as complementary to existing algorithmic work in DP, which focuses on designing methods that enjoy a smaller ε_{ub}.
- We develop an algorithmic framework for significantly improving ε_{lb}, as estimated by state-of-the-art audits in the hidden state, input space threat model (cf. Section 2). Our framework is a lightweight filtering-based wrapper (cf. Section 2), carefully designed so that, when applied to a range of popular DP-SGD variants, it retains the same provable ε_{ub}.

Empirical privacy measures are brittle [SM21, CCN+22, CJZ+22, AZT24], so reasoning about their guarantees takes care. In Section 4, we discuss the broader implications of ε_{lb} (2) as a definition. We observe here that targeting (2) opens an algorithmic design space that must go beyond DP-SGD. Indeed, despite a decade of research, no algorithm has dethroned the ε_{ub}-utility tradeoff of (Poisson subsampled) DP-SGD. This algorithm is also tightly auditable [NST+21, NHS+23, AC24, CBP25], with the notable caveat that existing audits have almost entirely targeted DP-SGD. In Section 3, we demonstrate that auditing our filtered variant with a comprehensive suite of tests from the literature yields $\varepsilon_{\text{lb}} \approx 0$, at a negligible utility drop from its unfiltered counterpart.

Although other strategies have been proposed for decreasing the auditability of ML algorithms (discussed in Section 1.3), our framework has a key qualitative advantage: it comes at *no loss* to the provable ε_{ub} of standard private optimizers. This is the focus of Appendix A, and making our proof flexible to a wide range of filters and optimizers is a main technical contribution of our work. For example, while our framework is related to robust statistics-inspired filters from the data poisoning literature [TLM18, HKSO21], these defenses break down when the fraction of poisoned points is too small (e.g., $< 1\%$). This is at odds with the auditing regime, where even a single canary can yield nearly-tight audits. Moreover, the filtering rules of [TLM18, HKSO21] were based on aggregate statistics (e.g., PCA). This makes it challenging to reason about privacy in datasets undergoing adaptive filtering, which intuitively could break the neighboring property over time.

Our privacy proof bypasses both issues, by restricting our filtering rules to be based on what we call *sample signatures*, defined in Section 2. Intuitively, these rules score individual samples in a way that only depends on other samples through a joint privately-trained model (so aggregate statistics such as the sample covariance of gradients cannot be used). We give a simple yet flexible proof that neighboring datasets that are filtered using sample signatures remain neighboring (Lemma 3), and that adaptive composition proofs are robust to this interleaved filtering (Lemma 2). Putting together these observations, we prove a privacy guarantee for our filtered DP-SGD method in Corollary 1 that matches Poisson-subsampled DP-SGD (Theorem 1). Section 3.1 then demonstrates that sample signatures are powerful enough to reliably filter input space canaries, substantially improving the Pareto frontier of utility-ε_{lb} tradeoffs against this test suite.

Of course, the metric (2) is only meaningful if the audit suite used is sufficiently representative of real-world attacks. Appendix E provides a systematic evaluation against all proposed input space canaries we are aware of, and Section 3.2 constructs canaries deliberately targeting our framework. Across all of these constructions, our framework yields negligible ε_{lb} values. In Appendix C, we critically examine potential vulnerabilities of our framework under a stronger (and oftentimes, unrealistic) *gradient space* threat model. Notably, the trainer can often enforce a particular threat model: for example, by computing their own gradients, and not publishing checkpoints, a practitioner can restrict training to the hidden state, input space setting.

1.3 Prior work

To our knowledge, no work has previously proposed the goal of lowering the auditable ε_{lb} (2) without compromising either the provable ε_{ub} or utility of a private learning algorithm. Here, we survey two lines of works that have aimed to balance two of these three criteria.

Empirically-private algorithms. Empirical privacy is often framed through membership inference, and indeed, a line of recent work from the security literature designs heuristics meant to protect against membership inference attacks (MIAs) [NSH18, JSB+19, SH21, CYF22, TMS+22, CP24]. These defenses aim at directly improving tradeoffs between the auditable ε_{lb} and the model utility, but do not consider provable privacy. Several of these works modify the training procedure or loss directly in a highly dataset-dependent way [NSH18, JSB+19, CYF22, TMS+22, CP24], which makes it difficult to perform theoretical privacy accounting. Other strategies are based on distillation from a

teacher model [SH21] and are therefore less flexible to setups where such a model is not available. We complement this literature by designing a heuristic defense with a privacy proof (Appendix A), showing it maintains a (lossless) ε_{ub}-utility tradeoff, competitive with the state-of-the-art. This modification makes measuring gaps between ε_{lb} and ε_{ub} meaningful.

Provable alternatives to DP-SGD. The DP literature has long considered tradeoffs between provable ε_{ub} and utility to be the gold standard for evaluating private learning algorithms. Alternatives to DP-SGD have been proposed, including PATE [PAE+17] (which uses public data), and augmentations of DP-SGD [ATMR21, DLC+21, TSL24] that adaptively estimate hyperparameters.

For simplicity, we focus on (subsampled variants of) DP-SGD as our baseline private optimizer of choice, as it remains an industry gold standard in privately training modern ML models [DBH+22]. While other private optimizers have gained traction recently as competitive alternatives, such as DP-FTRL [KMS+21, PUCC+25], work on the best way to audit models trained via alternative optimizers, a prerequisite to producing meaningful ε_{lb} estimates, is comparatively sparse. We do note that our framework does not currently apply to DP-FTRL, because that algorithm is not permutation-invariant (see discussion in Appendix A). We leave constructing a broader, provably private, filtering framework beyond permutation-invariant algorithms as an interesting open problem.

Finally, while we are not aware of prior works whose goal is explicitly to improve ε_{ub}-ε_{lb}-utility tradeoffs, [JUO20] observed that certain modifications to training (e.g., randomized initialization) result in better ε_{lb}, in a similar spirit to our motivation.

2 Overview

Privacy. For a sample space $\mathcal{S}$, we use $\mathcal{S}^* := \bigcup_{n\in\mathbb{N}} \mathcal{S}^n$ to denote an arbitrary-sized dataset from $\mathcal{S}$.

Definition 1 ((ε,δ)-DP)**.** *Let $(\varepsilon,\delta) \in \mathbb{R}_{\geq 0} \times [0,1]$. We say algorithm $\mathcal{A} : \mathcal{S}^* \to \Omega$ satisfies (ε,δ)-differential privacy (or, is (ε,δ)-DP) if for all events $\mathcal{E} \subseteq \Omega$, and all neighboring $\mathcal{D},\mathcal{D}' \in \mathcal{S}^*$,*

$$\Pr\left[\mathcal{A}(\mathcal{D}) \in \mathcal{E}\right] \leq \exp(\varepsilon)\Pr\left[\mathcal{A}(\mathcal{D}') \in \mathcal{E}\right] + \delta.$$

One subtlety in Definition 1 is that the literature uses slightly different definitions of *neighboring*; these notions of DP often imply each other, but only up to a constant factor loss. As our paper's methods are based on Poisson-subsampled DP-SGD, the de facto method for training provably-private ML models,[4] we use the definition admitting the tightest analyses of this algorithm, for fair comparison. Concretely, we use the add/remove definition of DP, where datasets $\mathcal{D} \in \mathcal{S}^n$, $\mathcal{D}' \in \mathcal{S}^m$ are *neighboring* if $|m-n| = 1$, and $|\mathcal{D} \setminus \mathcal{D}'| + |\mathcal{D}' \setminus \mathcal{D}| = 1$, i.e., they differ in one element.

The privacy analysis of DP-SGD is often performed using the following alternative definition.

Definition 2 (RDP)**.** *Let $n \in \mathbb{N}$, $\alpha > 1$, and $\rho \geq 0$. We say that $\mathcal{A} : \mathcal{S}^* \to \Omega$ satisfies (α,ρ)-RDP if for all neighboring datasets $\mathcal{D},\mathcal{D}' \in \mathcal{S}^*$, $D_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')) \leq \alpha\rho$, where D_α is the Rényi divergence and defined as $D_\alpha(P\|Q) = \frac{1}{\alpha-1}\log \mathbb{E}_{\omega\sim P}[(\frac{P(\omega)}{Q(\omega)})^{\alpha-1}]$.*

Lemma 1 gives a conversion from RDP to DP. For more properties of RDP, we refer to [Mir17].

Lemma 1 ([Mir17])**.** *If $\mathcal{M}$ satisfies (α,ε)-RDP, it satisfies $(\varepsilon + \frac{1}{\alpha-1}\log\frac{1}{\delta},\delta)$-DP for all $\delta \in (0,1)$.*

DP-SGD. Algorithm 3 outlines a typical Poisson-subsampled DP-SGD framework; for space considerations, we defer it to Appendix A, but highlight it as the black lines in Algorithm 5, which also highlights in red our modifications to basic DP-SGD. Algorithm 3 is used to train a model with parameters θ by minimizing the empirical loss $\frac{1}{n}\sum_{(x,y)\in D}\ell(\theta;x,y)$, where $\mathcal{D}$ is the size-n dataset. At each step of DP-SGD, we first (randomly) subsample a minibatch at rate q; then we compute the per-sample gradient $\nabla_\theta\ell(\theta;x,y)$ for each (x,y) within the batch $\mathcal{B}$ and clip their ℓ_2 norms to C; lastly, we average the clipped batch and add σ-scaled Gaussian noise to ensure privacy.

Computing the tightest upper bound of overall privacy cost of training a model is extensively studied in DP. Various accounting methods have been proposed to tackle this problem, e.g., moments accountant in [ACG+16], RDP in [MTZ19, ZW19], and PLD in [KJH20]. We recall the accounting method

[4] For example, privacy accounting via Poisson subsampling is the standard implementation in popular privacy packages such as Pytorch Opacus [YSS+21] and Tensorflow Privacy [Goo19].

based on RDP in Theorem 1, which gives a clean theoretical bound, implemented by popular privacy packages [YSS+21, Goo19]. We mention that PLD [KJH20] can achieve tighter ε_{ub} estimates in some cases than Theorem 1, but uses a numerical method that does not yield an explicit bound; our defense would enjoy an equally-tighter upper bound, as discussed in Appendix A.

Theorem 1 ([MTZ19]). *If $q < \frac{1}{5}, \gamma = \frac{\sigma C}{|\mathcal{B}|} > 4$, and $\alpha \leq \frac{\frac{1}{2}\gamma^2 L^2 - \ln 5 - 2\ln\gamma}{L + \ln(q\alpha) + (2\gamma^2)^{-1}}$ where $L := \log(1 + \frac{1}{q(\alpha-1)})$, Algorithm 3 is $(\alpha, T \cdot \frac{2q^2\alpha}{\sigma})$-RDP. It is also $(T \cdot \frac{2q^2\alpha}{\sigma} + \frac{1}{\alpha-1}\log\frac{1}{\delta}, \delta)$-DP for all $\delta \in (0, 1)$.*

Due to practical considerations, alternatives to Algorithm 3 are sometimes used, despite their weaker provable privacy-utility tradeoffs. For example, a common heuristic is to divide the dataset into randomly-shuffled minibatches of size m (see Algorithm 4) for N epochs; due to implementation details, this strategy often results in improved practical performance. Despite Theorem 1 not formally applying to this alternative algorithm, it is a common practice to apply its bound with the "in-expectation" equivalent parameters $q = \frac{m}{n}$, $T = N \cdot \lceil \frac{n}{m} \rceil$ to conclude a heuristic bound [DBH+22, PHK+23, CGK+24, LRKS25]. We sidestep this issue by directly using Algorithm 3 in our experiments, with its provable guarantee in Theorem 1. In Appendix A, we discuss how our privacy accounting for our filtered optimizers applies to Algorithm 4 as well. We also explain how our analysis framework extends to other recently-proposed alternatives, such as [CGH+25].

Threat models. The threat model specifies what information is known to the attacker/auditor and what strategies are allowed when constructing attacks. The literature on privacy auditing often uses "white-box" and "black-box" inconsistently as descriptors of threat models.

To sidestep this ambiguity, in this paper, we describe threat models along two main axes. The first axis is auditor access: in a *full state* threat model, the auditor has access to internal training information such as model parameters throughout training (e.g., intermediate checkpoints), while in a *hidden state* setting, intermediate training states are not revealed and the auditor only observes the final trained model (i.e., its weights). The second axis is what the auditor is allowed to perturb when designing an attack: *input space* attacks construct canaries by modifying the raw input (e.g., pixel values), while *gradient space* attacks operate by targeting the gradient signal a canary would produce during training (rather than directly optimizing the input itself).

Recent works [CBP25, BLIW26] suggest that the hidden state, input space threat model most accurately reflects the practice of private ML. While this setting is our focus, in Appendix C, we critically examine our framework's vulnerability to gradient space attacks accessing model history, and the reproducibility of these attacks in input space settings.

Framework. As discussed in Section 1.2, our defense is a wrapper that periodically filters training samples based on certain scoring rules. We provide pseudocode for the Poisson subsampled DP-SGD variant of our defense in Algorithm 1; we show how to bound its provable privacy in Corollary 1.

Sample signatures. A *sample signature* (or *scoring function*) maps samples to real numbers, based solely on information derived from that sample and privatized model parameters. Crucially, a sample signature does not depend on any other samples, except through the learned private model, in contrast to previous "batch" filtering rules proposed in the data poisoning literature [TLM18, HKSO21], which depended on e.g., the mean or covariance of the dataset gradients (a batch statistic). Use of sample signatures is essential for preserving privacy, an argument we formalize in Appendix A via a generalized composition theorem robust to sample signature filtering.

We denote a sample signature by $f_\theta(x, y)$ when scoring a sample (x, y), where the subscript indicates implicit dependence on the model. We consider several choices of f_θ; a few examples follow.

- Gradient norm: $f_\theta(x, y) := \|\nabla_\theta \ell(\theta; x, y)\|_p$ where, e.g., $p \in \{1, 2, \infty\}$. High scores suggest an influential sample in the trajectory of the model updates.
- Prediction margin: Let the (softmaxed) logits of example (x, y) with respect to model θ be denoted $p_\theta(x)[c]$ for a class c. Then we define $f_\theta(x, y) := -(p_\theta(x)[y] - \max_{c \neq y} p_\theta(x)[c])$. High scores suggest a sample with an unusual or misclassified label.

In Appendix F, we introduce several additional sample signatures, and provide an ablation study on the ability of these various scores to detect samples with high risk of memorization.

One major benefit of our outlier detection strategy is that it can detect outliers with significantly less signal than the "batch" update rules used by [TLM18, HKSO21], which state they require a constant

Algorithm 1: FilteredDPSGDPoisson$(\mathcal{D}, \theta_0, q, \eta, C, \sigma, T, m, f, k, \texttt{scope})$

Input: Dataset $\mathcal{D}$, model initialization θ_0, sample rate q, step size η, clip bound C, noise multiplier σ, step count T, epoch parameter m, sample signature f, selection parameter k, flag $\texttt{scope} \in \{\texttt{global}, \texttt{class}\}$

Output: Final model parameters θ

$(\mathcal{F}, S, \theta) \leftarrow (\emptyset, \{0\}^{|\mathcal{D}|}, \theta_0)$
for $t \in [T]$ **do**
  $\mathcal{B} \leftarrow \{\}$
  for $(x, y) \in \mathcal{D}$ **do**
    $r \leftarrow \text{Bern}(q)$
    if $r = 1$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup (x, y)$
    end
  end
  foreach $(x_i, y_i) \in \mathcal{B}$ **do**
    $g_i \leftarrow \begin{cases} \text{clip}(\nabla_\theta \ell(\theta; x_i, y_i), C) & i \notin \mathcal{F} \\ 0 & i \in \mathcal{F} \end{cases}$
    $S[i] \leftarrow f_\theta(x_i, y_i)$
  end
  $\xi \sim \mathcal{N}(0, (\frac{\sigma C}{|\mathcal{B}|})^2 I)$
  $\theta \leftarrow \theta - \eta(\frac{1}{|\mathcal{B}|} \sum_{(x_i, y_i) \in \mathcal{B}} g_i + \xi)$
  if $m \mid t$ **then**
    // Wait m minibatch iterations before filtering.
    $\mathcal{F} \leftarrow \mathcal{F} \cup \text{SelectTop}(S, \mathcal{D}, \mathcal{F}, k, \texttt{scope})$ // See Algorithm 2.
  end
end
return θ

fraction (e.g., 1%) of poisoned data to be effective. In DP auditing settings, this distinction is crucial, as memorization of even a single canary can nullify privacy bounds.

2.1 Empirical privacy reporting

We evaluate five (dataset, architecture) pairs: MNIST [LBBH98] with a CNN; CIFAR-10 [Kri09] with a CNN, a CNN with augmentation multiplicity, and a WideResNet-16 (WRN-16); and Purchase100 [SSSS17] with an MLP. The CNN and MLP models use Xavier initialization [GB10], and the WideResNet-16 uses Kaiming initialization [HZRS15]. Purchase100 is a common benchmark for membership inference, but prior work notes that its simplicity limits the generalizable conclusions it can support [CCN+22]. Within each audit, all models share the same initialization but are trained with distinct random seeds, yielding independent trials as in [AC24]. Unless otherwise specified, we use $\varepsilon_{\text{ub}} = 10$, and our defense scores samples by their clipped gradient L_∞ norm, filters per class, and removes 5 samples per class per epoch (1 per class for Purchase100). Each audit uses 400 models: 200 trained with the canary and 200 trained without it.

Following [NHS+23], each audit records every model's loss on the canary and selects the threshold that best separates models trained with the canary from those trained without it. We use this threshold to estimate the false positive rate (FPR) and false negative rate (FNR), and apply the Clopper-Pearson method to obtain conservative upper confidence bounds on both error rates. These bounds are then converted into a lower bound ε_{lb} on the privacy parameter via (6). We defer a more detailed discussion of our auditing methodology to Appendix B, including our experimental setup (Appendix B.1).

2.2 Privacy of Algorithm 1

In Appendix A.1, we show how to obtain the following privacy guarantee on Algorithm 1.

Corollary 1. *Algorithm 1 satisfies the same privacy bound as stated in Theorem 1 for Algorithm 3.*

Corollary 1 arises as a consequence of a more general framework we develop in Appendix A.1, for reasoning about privacy preservation with interleaved filtering steps. We state the two main lemmas used in this framework here, with extended discussions deferred to Appendix A.1.

Our first helper lemma states that Rényi DP composition bounds persist, even if in between each private release, we replace a pair of adjacent datasets with new datasets that remain adjacent.

Lemma 2 (Interleaved Rényi composition)**.** *Let $\mathcal{A}_1 : \mathcal{S}^* \to \Omega_1$ be (α, ρ_1)-RDP, and let $\mathcal{A}_2 : \mathcal{S}^* \times \Omega_1 \to \Omega_2$ be such that $\mathcal{A}_2(\cdot, \omega_1)$ is (α, ρ_2)-RDP for any $\omega_1 \in \Omega_1$. Also, let $\mathcal{F} : \mathcal{S}^* \times \Omega_1 \to \mathcal{S}^*$ be a* filter *such that for any neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^*$, we always have that $\mathcal{F}(\mathcal{D}, \omega_1)$ and $\mathcal{F}(\mathcal{D}', \omega_1)$ are neighboring for any $\omega_1 \in \Omega_1$. Then, the mechanism $\mathcal{A}$ that first samples $\omega_1 \leftarrow \mathcal{A}_1(\mathcal{D})$ and then samples $\omega_2 \leftarrow \mathcal{A}_2(\mathcal{F}(\mathcal{D}, \omega_1), \omega_1)$ (i.e., applies $\mathcal{F}$ to $\mathcal{D}$ before $\mathcal{A}_2$) is $(\alpha, \rho_1 + \rho_2)$-RDP.*

Our second helper lemma states that, up to permutation, two datasets modified using top-k filtering via *sample signatures* cannot have a numbering of neighboring elements that grows. This only preserves (ordered) neighboring for permutation-invariant algorithms, which include subsampled and shuffled variants of DP-SGD, as well as other private algorithms discussed in Appendix A.2.

Lemma 3. *Let D_1, D_2 be multisets of real numbers, and let $k \leq \min\{|D_1|, |D_2|\}$. Let F_1 be D_1 with its k largest elements removed, and define F_2 similarly with respect to D_2. Then,*

$$|D_1 \setminus D_2| + |D_2 \setminus D_1| \geq |F_1 \setminus F_2| + |F_2 \setminus F_1|. \tag{3}$$

3 Privacy-Preserving Empirically Private Learning

In this section, we present experiments outlining the performance of our filtered algorithms in reducing ε_{lb} for DP-SGD across standard vision and tabular datasets.

3.1 Input space attacks

Input space attacks operate at the data level by injecting carefully crafted training samples into the dataset. We first evaluate our defense on a singular canary across varying privacy budgets ($\varepsilon \in \{2, 4, 6, 8, 10\}$), datasets, model architectures, and training configurations ("AM4" refers to training with augmentation multiplicity using 4 augmentations). For MNIST and CIFAR-10, we use a blank (zero) canary; for Purchase100, we use a random dense vector canary, as a blank input to an MLP propagates no signal through the network (the hidden states are driven entirely by the biases), and is therefore not auditable.

We find that our defense generally reduces ε_{lb} with minimal impact on model utility. Notably, our defense reduced the ε_{lb} for MNIST on a CNN and CIFAR-10 on a WRN-16 (the model with the test utility on a more "complex" dataset) to ≈ 0 for almost every setting of $\varepsilon_{\text{ub}} \in \{2, 4, 6, 8, 10\}$ even after applying both heuristics in Appendix B. Moreover, we found that alternative scoring rules detailed in Appendix F, including prediction entropy and gradient kurtosis, substantially reduce the auditable ε_{lb} for several of the other dataset / model combinations.

Our defense may not universally lower ε_{lb}; MNIST and CIFAR-10 on WRN-16 at $\varepsilon_{\text{ub}} = 2$ and Purchase at $\varepsilon_{\text{ub}} = 6$ observe an increase in ε_{lb} post-defense. We believe that this apparent discrepancy may potentially arise from finite sample error. In particular, the 90% two-sided confidence intervals overlap for MNIST at $\varepsilon_{\text{ub}} = 2$ and Purchase at $\varepsilon_{\text{ub}} = 6$, so this could just be an artifact of the number of models used. The 90% confidence interval for CIFAR-10 at $\varepsilon_{\text{ub}} = 2$ without the defense was not very informative, as it was $(0.0, 0.0)$, but the ε_{lb} with the defense was also negligible in this case. Notably, the reversal cases identified either had a small ε_{lb} to begin with, or were on the Purchase dataset, previously remarked on as potentially ill-suited for privacy audits in [CCN+22].

While the accuracy gaps are generally contained to ± 1, notice the anomalous and relatively large accuracy gap for the Purchase dataset post-defense at $\varepsilon_{\text{ub}} = 10$. In general, we believe that our finding that the accuracy was minimally affected is fairly robust: on all 24 other tasks, both training and test accuracy dropped by 1.5% or less. For the specific Purchase at $\varepsilon_{\text{ub}} = 10$ task, we believe the steeper accuracy drop may be due to the simplicity of the dataset (as highlighted in [CCN+22]), or due to use of a high ε_{ub} level causing more sensitivity to dropping a small amount of data.

We also evaluate our defense against harder input space constructions that better reflect membership inference threats. We test three attack variants on CIFAR-10 and MNIST at ε_{ub} = 10 (Table 2).

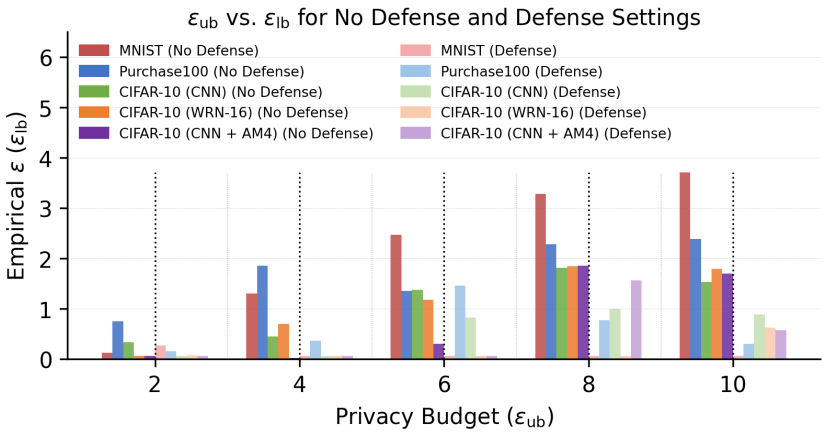


Figure 1: ε_{lb} across varying ε_{ub} budgets.

Dataset (Model)	**Theoretical ε**	**Train Acc (%)**	**Test Acc (%)**
MNIST (CNN)	2	97.47/96.36	97.40/96.57
MNIST (CNN)	4	98.42/97.23	98.37/97.52
MNIST (CNN)	6	98.65/97.44	98.53/97.74
MNIST (CNN)	8	98.75/97.54	98.61/97.86
MNIST (CNN)	10	98.80/97.59	98.64/97.89
Purchase (MLP)	2	77.82/76.79	74.51/73.19
Purchase (MLP)	4	87.74/87.74	84.09/84.09
Purchase (MLP)	6	89.96/90.08	86.92/86.91
Purchase (MLP)	8	90.40/90.40	87.12/87.12
Purchase (MLP)	10	90.57/85.98	87.69/83.11
CIFAR-10 (CNN)	2	32.89/32.77	32.81/32.87
CIFAR-10 (CNN)	4	61.28/60.26	59.33/58.57
CIFAR-10 (CNN)	6	70.14/68.59	66.93/65.55
CIFAR-10 (CNN)	8	73.97/72.91	69.56/69.05
CIFAR-10 (CNN)	10	76.15/75.35	71.18/70.85
CIFAR-10 (WideResNet-16)	2	53.14/54.07	52.53/53.17
CIFAR-10 (WideResNet-16)	4	62.86/63.60	61.75/62.42
CIFAR-10 (WideResNet-16)	6	68.40/68.63	66.57/67.07
CIFAR-10 (WideResNet-16)	8	72.40/70.99	70.13/68.75
CIFAR-10 (WideResNet-16)	10	73.87/72.97	71.36/70.70
CIFAR-10 (CNN + AM4)	2	31.28/31.45	31.72/31.45
CIFAR-10 (CNN + AM4)	4	57.05/56.36	56.36/56.36
CIFAR-10 (CNN + AM4)	6	66.07/65.26	65.24/64.46
CIFAR-10 (CNN + AM4)	8	69.66/69.16	68.36/68.36
CIFAR-10 (CNN + AM4)	10	71.52/70.94	70.49/70.03

Table 1: Defense performance on private models across privacy budgets $\varepsilon \in \{2, 4, 6, 8, 10\}$. Training and testing accuracies are reported in the table as *No Defense/Defense*.

First, we used adversarial FGSM examples, which iteratively apply small bounded perturbations to a sample until the model predicts a target class [GSS15]. Second, we tested mislabeled in-distribution images, where random training samples are assigned incorrect labels uniformly at random from the remaining classes. Third, we implemented ClipBKD, which identifies the direction of least variance in the training data distribution via singular value decomposition and constructs a canary along the smallest singular vector [JUO20]. Across all attacks, our defense reduced empirical privacy loss. We note that Appendix E contains further experiments on multi-canary audits.

Attack	Defense Setting	Dataset (Model)	Train Acc (%)	Test Acc (%)	Empirical ε
FGSM	No Defense	CIFAR-10 (CNN)	76.06	70.75	1.14
FGSM	Defense	CIFAR-10 (CNN)	74.95	70.14	0.07
FGSM	No Defense	MNIST (CNN)	98.78	98.57	3.03
FGSM	Defense	MNIST (CNN)	97.58	97.78	0
Mislabeled	No Defense	CIFAR-10 (CNN)	76.31	71.37	0.47
Mislabeled	Defense	CIFAR-10 (CNN)	75.05	70.61	0
Mislabeled	No Defense	MNIST (CNN)	98.80	98.64	2.45
Mislabeled	Defense	MNIST (CNN)	97.59	97.89	0.69
ClipBKD	No Defense	CIFAR-10 (CNN)	76.13	71.02	2.93
ClipBKD	Defense	CIFAR-10 (CNN)	75.74	70.86	0
ClipBKD	No Defense	MNIST (CNN)	98.88	98.60	3.46
ClipBKD	Defense	MNIST (CNN)	97.59	97.89	0

Table 2: Empirical privacy loss under non-blank input space attacks for $\varepsilon_{\text{ub}} = 10$.

3.2 Defense-aware input space attack

Beyond evaluating previously-proposed input space canaries in the literature (in Section 3.1), here we detail an attempt to learn a defense-aware canary that dodges our filter while remaining auditable. We formulate this as a bi-level optimization problem, where the inner loop simulates the training procedure and the outer loop updates the canary.

Let x denote a candidate canary; we want to map x to y_w, a target incorrect label. For a given x, the inner loop trains the model using DP-SGD and our gradient-norm filtering defense. Let $K(x)$ denote the epoch at which the defense detects x, with $K(x) = -1$ if the canary is never detected. When the canary is detected on epoch $K(x)$, let $\theta_{K(x)}$ denote the model parameters used to compute the defense score on that epoch, and let $\tau_{K(x)}$ denote the corresponding detection threshold i.e. the k-th largest gradient score in that epoch.

$$\mathcal{L}(x) = \ell\big(f_{\theta_0}(x),\, y_w\big) + \lambda \cdot \begin{cases} \max\Big(0,\ \big\|\nabla_\theta \ell\big(f_{\theta_{K(x)}}(x),\, y_w\big)\big\|_\infty - \tau_{K(x)}\Big), & \text{if } K(x) \neq -1, \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The first term of (4) pushes the canary to be compatible with its assigned label under the initial model. We use this initial-model loss as a tractable proxy for auditability because directly optimizing the canary's loss after training would require differentiating through the entire DP-SGD training trajectory. Using the loss at the detection epoch suffers from the same problem and would also create a dependency on $K(x)$. The model initialization, θ_0, is set independent of the canary, so this offers a simple differentiable objective with respect to x.

The second term of (4) penalizes the canary when it has a gradient above the detection threshold, with a penalty proportional to its distance above the detection threshold.

The λ parameter controls the trade-off between auditability and defense evasion.

The canary is updated using an FGSM-style step. For our experiments, we optimize a canary chosen from class 1 and assigned a target label 0. We use 50 outer optimization steps with a fixed step size of 1e-2 and 3 independent inner training runs per outer step.

Table 3 shows that across various λ, while we increase the average epoch on which the canary is dropped, we cannot push the drop epoch past the end of training (100 epochs). We run a full audit on the learned canary with the largest drop epoch, but we find that the initial canary (pre-optimization) yields a higher ε_{lb} than our learned canary (2.453 vs 1.962). Post defense, the initial canary's ε_{lb} drops to 0.65, while our learned canary's ε_{lb} drops to 0.

4 Conclusion

In this conclusion, we summarize our main contributions and findings, discuss practical considerations when using our methods, and provide a roadmap to our appendices.

λ	0.0	0.0	0.1	0.5	1.0	5.0	10.0	100.0
Average drop epoch	5.7	6.0	16.7	14.0	15.0	13.0	13.0	14.0

Table 3: Defense-aware canary optimization attempts across different values of λ.

Contributions and findings. In this work, we revisit the apparent near-tightness between the auditable and provable privacy of DP-SGD [NHS+23, AC24], from an algorithm design perspective. We propose using the auditable privacy ε_{lb} as a metric to evaluate private learning algorithms, complementary to the utility and provable privacy ε_{ub}. A key qualitative advantage of ε_{lb} is its threat model-awareness: whereas standard composition-based privacy proofs do not capture deployment-specific restrictions on auditors, ε_{lb} can be evaluated with a restricted audit suite appropriate for a particular deployment. We then develop a filtering-based wrapper around DP-SGD, that losslessly retains the ε_{ub} of any composition-based privacy proof for a permutation-invariant algorithm.

Our main empirical finding is that our filter substantially improves ε_{lb} relative to DP-SGD, in the input space, hidden state threat model, often viewed as the most reflective of practical deployments [YS22, CBP25, BLIW26]. Across a broad suite of input space, hidden state audits, our framework typically reduces ε_{lb} by a large constant factor, or drives it to zero (Figure 1, Table 2). The few reversals occur only when DP-SGD already achieved a near-zero ε_{lb}, or when the dataset used was Purchase, known to be a simpler target for audits [CCN+22]. Moreover, in 24 of 25 configurations, including all that did not use Purchase, the test-time utility dropped by 1% or less (Tables 1 and 2). These findings continue to hold for adaptive canaries we constructed to target our defense (Table 3).

The implications of ε_{lb} as a metric. Our proposed ε_{lb} is not a formal security guarantee against all possible privacy attacks, and thus should be interpreted as measuring the vulnerability of an ML model against a current suite of representative attacks. These attacks' efficacy is measured by the same distinguishing task as in (2), and thus it is reasonable to base our test suite off of state-of-the-art audits. As the test suite in (2) expands with future work, ε_{lb} becomes a "moving target" by nature. Notably, any newly-developed auditing strategies can only strengthen the value of this target.

Our hope is that formalizing ε_{lb} as an evaluation metric initiates a back-and-forth between auditing researchers aiming to expose privacy vulnerabilities, and algorithm designers aiming to withstand existing audits. Indeed, our evaluations reveal a blind spot of current input space audits: either our filtered algorithms truly have better privacy than the theory predicts, or new tests must be developed to accurately audit their privacy. Without ε_{lb} as an evaluation metric, or assessing audit designs beyond DP-SGD, there would be no impetus for developing stronger tests. Finally, if a persistent gap between provable and auditable privacy is robustly identified in a particular threat model, this creates an opportunity for theorists to formalize the underlying privacy guarantee.

Further practical considerations. Our appendix sections provide evaluations of two practical considerations: robustness to the threat model, and impact on non-canary samples.

Regarding the former, Appendix C examines a substantially stronger gradient space, multi-canary threat model, and shows that coordinated attacks with knowledge of our hyperparameters can successfully exploit our defense, although the resulting ε_{lb} is a measure of group privacy. We also attempt to reproduce this attack in our input space setting, but find that the natural constructions fail. Notably, in practice, a model trainer can often enforce the hidden state, input space threat model by withholding intermediate checkpoints.

Regarding the latter, the general tension between privacy and fairness is a known phenomenon [BPS19, UNK+21] that is out of our scope to address in full. Nevertheless, Appendix D shows that our framework's disparate impact is mild when minority groups are sufficiently large, or few points are filtered. In Appendix I, we also examine the "privacy onion" effect (i.e., whether our defense exposes other samples beyond the planted canary), which prior works on evaluating empirical privacy have cautioned against [CJZ+22]. We demonstrate empirically that even the strongest adaptively selected non-canary attack on our filtered defense is substantially weaker than the original canary attack on DP-SGD, a gap widened after controlling for false discovery.

Finally, Appendices A and B contain deferred discussions and supporting results on our privacy proof and auditing methodology, Appendices E and H provide further evaluations on input space canaries, and Appendices F, G, J, K, and L contain ablation studies demonstrating the robustness of our findings to hyperparameters and other design decisions.

Acknowledgments

KT thanks Jonathan Ullman and Florian Tramèr for helpful conversations during this project's conception. We thank Milad Nasr for clarifying conversations regarding the prior work [NHS+23], and both Milad Nasr and Thomas Steinke for providing feedback on drafts of this paper. We are grateful to the Texas Advanced Computing Center (TACC) and the UT Austin Center for Generative AI for providing the computing resources used in this project. GK was supported by a Canada CIFAR AI Chair, an NSERC Discovery Grant, and an Ontario Early Researcher Award.

References

[ABGL19] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

[AC24] Meenatchi Sundaram Muthu Selva Annamalai and Emiliano De Cristofaro. Nearly tight black-box auditing of differentially private machine learning. In *Advances in Neural Information Processing Systems 37*, NeurIPS '24, pages 131482–131502. Neural Information Processing Systems Foundation, Inc., 2024.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, CCS '16, pages 308–318. ACM, 2016.

[AT22] Jason M. Altschuler and Kunal Talwar. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

[ATMR21] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

[AZT24] Michael Aerni, Jie Zhang, and Florian Tramèr. Evaluations of machine learning privacy defenses are misleading. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1271–1284, 2024.

[BLIW26] Matteo Boglioni, Terrance Liu, Andrew Ilyas, and Zhiwei Steven Wu. Optimizing canaries for privacy auditing with metagradient descent. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*. OpenReview.net, 2026.

[BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 15479–15488. Curran Associates, Inc., 2019.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 464–473. IEEE Computer Society, 2014.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CBP25] Tudor Ioan Cebere, Aurélien Bellet, and Nicolas Papernot. Tighter privacy auditing of DP-SGD in the hidden state threat model. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net, 2025.

[CCN+22] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[CGH+25] Lynn Chua, Badih Ghazi, Charlie Harrison, Pritish Kamath, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Balls-and-bins sampling for DP-SGD. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2025*, volume 258 of *Proceedings of Machine Learning Research*, pages 946–954. PMLR, 2025.

[CGK+24] Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How private are DP-SGD implementations? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8904–8918. PMLR, 2024.

[CJZ+22] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. In *Advances in Neural Information Processing Systems 35*, NeurIPS '22, pages 13263–13276. Curran Associates, Inc., 2022.

[CP24] Zitao Chen and Karthik Pattabiraman. Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction. In *31st Annual Network and Distributed System Security Symposium, NDSS 2024*. The Internet Society, 2024.

[CTW+21] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, USENIX Security '21, pages 2633–2650. USENIX Association, 2021.

[CYF22] Dingfan Chen, Ning Yu, and Mario Fritz. Relaxloss: Defending membership inference attacks without losing utility. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net, 2022.

[DBH+22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

[DLC+21] Jian Du, Song Li, Xiangyi Chen, Siheng Chen, and Mingyi Hong. Dynamic differential-privacy preserving sgd. *arXiv preprint arXiv:2111.00173*, 2021.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DRS22] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60. IEEE Computer Society, 2010.

[FS25] Vitaly Feldman and Moshe Shenfeld. Privacy amplification by random allocation. In *Advances in Neural Information Processing Systems 38*, NeurIPS '25, 2025.

[FS26] Vitaly Feldman and Moshe Shenfeld. Efficient privacy loss accounting for subsampling and random allocation. In *Forty-third International Conference on Machine Learning*, 2026.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *AISTATS '10*, pages 249–256. JMLR, Inc., 2010.

[GKK+26] Juan Felipe Gomez, Bogdan Kulynych, Georgios Kaissis, Flavio du Pin Calmon, Jamie Hayes, Borja Balle, and Antti Honkela. Gaussian dp for reporting differential privacy guarantees in machine learning. In *2026 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, SaTML '26. IEEE, 2026.

[Goo19] Google. Tensorflow privacy. `https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy`, 2019.

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, ICLR '15, 2015.

[HKSO21] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Defense against backdoor attacks via robust covariance estimation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 4129–4139. PMLR, 2021.

[HSR+08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8):1–9, 2008.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, ICCV '15, pages 1026–1034. IEEE Computer Society, 2015.

[JSB+19] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019*, pages 259–274. ACM, 2019.

[JUO20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In *Advances in Neural Information Processing Systems 33*, NeurIPS '20. Curran Associates, Inc., 2020.

[KJH20] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2560–2569. PMLR, 2020.

[KMS+21] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *ICML '21*, pages 5213–5225. PMLR, 2021.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LRKS25] Christian Janos Lebeda, Matthew Regehr, Gautam Kamath, and Thomas Steinke. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition. In *Third IEEE Conference on Secure and Trustworthy Machine Learning*, SaTML '25. IEEE Computer Society, 2025.

[Mir17] Ilya Mironov. Rényi differential privacy. In *Proceedings of the 30th IEEE Computer Security Foundations Symposium*, CSF '17, pages 263–275. IEEE Computer Society, 2017.

[MSH+22] Shubhankar Mohapatra, Sajin Sasy, Xi He, Gautam Kamath, and Om Thakkar. The role of adaptive optimizers for honest private hyperparameter selection. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, volume 36 of *AAAI '22*, pages 7806–7813, 2022.

[MSS23] Samuel Maddock, Alexandre Sablayrolles, and Pierre Stock. CANIFE: Crafting canaries for empirical privacy measurement in federated learning. In *Proceedings of the 11th International Conference on Learning Representations*, ICLR '23, 2023.

[MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[NHS+23] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *32nd USENIX Security Symposium, USENIX Security 2023*, pages 1631–1648. USENIX Association, 2023.

[NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pages 634–646. ACM, 2018.

[NST+21] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, SP '21, pages 866–882. IEEE, 2021.

[PAE+17] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[PHK+23] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H. Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, July 2023.

[PS22] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

[PUCC+25] Krishna Pillutla, Jalaj Upadhyay, Christopher A Choquette-Choo, Krishnamurthy Dvijotham, Arun Ganesh, Monika Henzinger, Jonathan Katz, Ryan McKenna, H Brendan McMahan, Keith Rush, Thomas Steinke, and Abhradeep Thakurta. Correlated noise mechanisms for differentially private learning. *arXiv preprint arXiv:2506.08201*, 2025.

[Roc97] R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP '13, pages 245–248. IEEE Computer Society, 2013.

[SH21] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021*, pages 9549–9557. AAAI Press, 2021.

[SKHL20] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[SM21] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium, USENIX Security 2021*, pages 2615–2632. USENIX Association, 2021.

[SNJ23] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. In *Advances in Neural Information Processing Systems 36*, NeurIPS '23, pages 49268–49280. Curran Associates, Inc., 2023.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, SP '17, pages 3–18. IEEE Computer Society, 2017.

[TB21] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *Proceedings of the 9th International Conference on Learning Representations*, ICLR '21, 2021.

[TLM18] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pages 8011–8021. Curran Associates, Inc., 2018.

[TMS+22] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by self-distillation through a novel ensemble architecture. In *31st USENIX Security Symposium, USENIX Security 2022*, pages 1433–1450. USENIX Association, 2022.

[TSL24] Qiaoyue Tang, Frederick Shpilevskiy, and Mathias Lécuyer. Dp-adambc: Your dp-adam is actually dp-sgd (unless you apply bias correction). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15276–15283, 2024.

[UNK+21] Archit Uniyal, Rakshit Naidu, Sasikanth Kotti, Sahib Singh, Patrik Joslin Kenfack, Fatemehsadat Mireshghallah, and Andrew Trask. DP-SGD vs PATE: which has less disparate impact on model accuracy? *CoRR*, abs/2106.12576, 2021.

[WZ10] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[YS22] Jiayuan Ye and Reza Shokri. Differentially private learning needs hidden state (or much faster convergence). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

[YSS+21] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[ZBWT+23] Santiago Zanella-Beguelin, Lukas Wutschitz, Shruti Tople, Ahmed Salem, Victor Rühle, Andrew Paverd, Mohammad Naseri, Boris Köpf, and Daniel Jones. Bayesian estimation of differential privacy. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML '23*, pages 40624–40636. PMLR, 2023.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the 2016 British Machine Vision Conference*, BMVC '16, pages 87:1–87:12. BMVA Press, 2016.

[ZSZ+22] Michael Zhang, Nimit Sharad Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pages 26484–26516. PMLR, 2022.

[ZW19] Yuqing Zhu and Yu-Xiang Wang. Poisson subsampled rényi differential privacy. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7634–7642. PMLR, 2019.

A Privacy of Filtering via Sample Signatures

Here we provide a framework to analyze the privacy of augmenting a DP algorithm with filtering. The unfiltered variants of our algorithms, Algorithm 3 and 4, have two useful properties. First, their privacy is based on composition of individual steps' (Rényi) privacy. Second, the algorithms are permutation-invariant, in that the same sequence of random operations (in distribution) is performed even if we arbitrarily shuffle the data. These properties hold for many DP algorithms; the latter specifically is natural if assumptions placed on dataset samples are uniform. Our framework for arguing about privacy with interleaved filtering steps carefully uses these two properties.

A.1 Analysis

The first piece of our framework is a generalized privacy composition theorem. Intuitively, it states that if before we apply privacy composition, we can arbitrarily replace a pair of datasets with new datasets with the same number of neighbors, this cannot change the final privacy bound.

Lemma 2 (Interleaved Rényi composition)**.** *Let $\mathcal{A}_1 : \mathcal{S}^* \to \Omega_1$ be (α, ρ_1)-RDP, and let $\mathcal{A}_2 : \mathcal{S}^* \times \Omega_1 \to \Omega_2$ be such that $\mathcal{A}_2(\cdot, \omega_1)$ is (α, ρ_2)-RDP for any $\omega_1 \in \Omega_1$. Also, let $\mathcal{F} : \mathcal{S}^* \times \Omega_1 \to \mathcal{S}^*$ be a* filter *such that for any neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^*$, we always have that $\mathcal{F}(\mathcal{D}, \omega_1)$ and $\mathcal{F}(\mathcal{D}', \omega_1)$ are neighboring for any $\omega_1 \in \Omega_1$. Then, the mechanism $\mathcal{A}$ that first samples $\omega_1 \leftarrow \mathcal{A}_1(\mathcal{D})$ and then samples $\omega_2 \leftarrow \mathcal{A}_2(\mathcal{F}(\mathcal{D}, \omega_1), \omega_1)$ (i.e., applies $\mathcal{F}$ to $\mathcal{D}$ before $\mathcal{A}_2$) is $(\alpha, \rho_1 + \rho_2)$-RDP.*

Proof. We closely follow the proof of Proposition 1, [Mir17]. Fix a pair of outcomes $\omega = (\omega_1, \omega_2) \in \Omega_1 \times \Omega_2$. For simplicity, we overload notation so that $\mathcal{A}_1(\mathcal{D})(\omega_1)$ refers to the density of outcomes according to $\mathcal{A}_1(\mathcal{D})$, and define $\mathcal{A}_2(\mathcal{D}, \omega_1)(\omega_2)$, $\mathcal{A}(\mathcal{D})(\omega)$ similarly. Finally, we use the shorthand $\mathcal{D}_{\omega_1}$ to mean $\mathcal{F}(\mathcal{D}, \omega_1)$ for all $\omega_1 \in \Omega_1$. We have the desired

$$\exp\left((\alpha-1)D_\alpha\left(\mathcal{A}(\mathcal{D})\|\mathcal{A}(\mathcal{D}')\right)\right) = \int_{\omega\in\Omega_1\times\Omega_2}\left(\frac{\mathcal{A}(\mathcal{D})(\omega)}{\mathcal{A}(\mathcal{D}')(\omega)}\right)^\alpha \mathcal{A}(\mathcal{D}')(\omega)\mathrm{d}\omega$$

$$= \int_{\omega_1\in\Omega_1}\left(\frac{\mathcal{A}_1(\mathcal{D})(\omega_1)}{\mathcal{A}_1(\mathcal{D}')(\omega_1)}\right)^\alpha\left(\int_{\omega_2\in\Omega_2}\left(\frac{\mathcal{A}_2(\mathcal{D}_{\omega_1},\omega_1)(\omega_2)}{\mathcal{A}_2(\mathcal{D}'_{\omega_1},\omega_1)(\omega_2)}\right)^\alpha \mathcal{A}_2(\mathcal{D}'_{\omega_1},\omega_1)(\omega_2)\mathrm{d}\omega_2\right)\mathcal{A}_1(\mathcal{D}')(\omega_1)\mathrm{d}\omega_1$$

$$\le \exp((\alpha-1)\rho_2)\int_{\omega_1\in\Omega_1}\left(\frac{\mathcal{A}_1(\mathcal{D})(\omega_1)}{\mathcal{A}_1(\mathcal{D}')(\omega_1)}\right)^\alpha \mathcal{A}_1(\mathcal{D}')(\omega_1)\mathrm{d}\omega_1 \le \exp((\alpha-1)(\rho_1+\rho_2)).$$

The first inequality used the privacy assumption on $\mathcal{A}_2(\cdot, \omega_1)$, and that $\mathcal{D}_{\omega_1}$ and $\mathcal{D}'_{\omega_1}$ are neighboring, for any choice of ω_1. The second inequality used the privacy assumption on $\mathcal{A}_1$. □

The strategy in Lemma 2 is quite general, and we expect it to apply to many if not all settings where privacy composition holds. For example, it is straightforward to apply the strategy to prove generalized variants of basic and advanced composition under filtering (Theorems 3.16, 3.20, [DR14]). This strategy also works for composition of PLD curves, see e.g., Lemma C.3 of [DRS22], which also first conditions on the shared outcome of $\mathcal{A}_1$ before applying composition with $\mathcal{A}_2$.

Importantly in our applications to our filtered defense, if we are applying Lemma 2 sequentially (e.g., T times to compose T DP-SGD steps), ω_1 in the lemma statement encompasses the entire history of $T-1$ prior operations, so the operations $\mathcal{F}(\cdot, \omega_1)$ can depend on all previous iterations.

We next state the formal subroutine for selecting the top-scored samples to remove from our dataset.

The second piece of our framework is an observation that taking two neighboring datasets, scoring samples based on a coupled rule, and dropping the k largest-scored samples each, cannot increase the Hamming distance between the datasets. This observation treats dropped samples as equivalent (e.g., replaced with a standardized dummy sample), regardless of their indices. This is where we require permutation-invariance: otherwise (if indices are accounted for) the distance can grow.[5] Our proof in fact handles both main definitions of neighboring used in the DP literature, namely, the add/remove notion used in our work, as well as the replacement notion.

[5] An illustrative example is neighboring size-2 datasets with scores $\{1, 2\}$ and $\{3, 2\}$. Replacing the largest score with a common dummy causes 2-neighboring datasets, but they remain 1-neighboring if indices are ignored.

Algorithm 2: SelectTop$(S, \mathcal{D}, \mathcal{F}, k, \texttt{scope})$

Input: Computed sample signatures S, dataset $\mathcal{D}$, filtered set $\mathcal{F} \subseteq \mathcal{D}$, count $k \in \mathbb{N}$, mode $\texttt{scope} \in \{\texttt{global}, \texttt{class}\}$

Output: Selected samples $\mathcal{T}$

$\mathcal{U} \leftarrow \mathcal{D} \setminus \mathcal{F}$

if *scope* $=$ *global* **then**

  return k *indices* $i \in \mathcal{U}$ *with largest* $S[i]$

else

  foreach *class* c *represented in* $\mathcal{U}$ **do**

    $\mathcal{T}_c \leftarrow k$ indices $i \in \mathcal{U}_c := \{i \in \mathcal{U} \mid y_i = c\}$ with largest $S[i]$

  end

  return $\bigcup_{\text{class } c \text{ represented in } \mathcal{U}} \mathcal{T}_c$

end

Lemma 3. *Let D_1, D_2 be multisets of real numbers, and let $k \leq \min\{|D_1|, |D_2|\}$. Let F_1 be D_1 with its k largest elements removed, and define F_2 similarly with respect to D_2. Then,*

$$|D_1 \setminus D_2| + |D_2 \setminus D_1| \geq |F_1 \setminus F_2| + |F_2 \setminus F_1|. \tag{3}$$

Proof. Denote $s := |D_1 \cap D_2|$ and $t := |F_1 \cap F_2|$. Because

$$|D_1 \setminus D_2| + |D_2 \setminus D_1| = (|D_1| - s) + (|D_2| - s) = |D_1| + |D_2| - 2s,$$

and similarly the right-hand side of (3) is $|D_1| + |D_2| - 2t - 2k$, our goal in (3) is equivalent to showing that $t \geq s - k$. If $s \leq k$ then this is clear. Assume $s > k$ henceforth.

Sort $D_1 \cap D_2$ in order: $x_1 \geq x_2 \geq \ldots \geq x_s$. Since $s > k$, then for any $j > k$, we must have $x_j \in F_1 \cap F_2$ because we remove the k highest scores of each set only (coupling any copies of duplicated items removed). Thus at least $s - k$ common elements survive and $t \geq s - k$ as claimed. □

Notice that the condition (3) captures neighboring-preservation under filtering, up to assigning indices to the elements (i.e., the locations of the removed elements do not matter). This last caveat explains why our framework requires permutation-invariance of the algorithm. Indeed, under the add/remove definition of neighboring, the two sides of (3) respectively equal the number of add/remove operations needed to change one set into the other, before and after filtering. Similarly, under the replacement notion of DP (Definition 1, [DMNS06]), the two sides of (3) are both twice the number of replacement operations needed for this transformation.

By combining Lemma 2 and 3, we can conclude that any permutation-invariant algorithm whose privacy analysis is based on RDP composition retains the same privacy bound under arbitrary sample signature-based filtering. The fact that filtering is performed based on sample signatures (so that the score of each sample is the same as long as the model is coupled) is critical in applying Lemma 3. As an example application, we analyze the privacy of Algorithm 1, compared to its unfiltered variant (i.e., the lines in Algorithm 1 not marked in red), which we reproduce here.

Corollary 1. *Algorithm 1 satisfies the same privacy bound as stated in Theorem 1 for Algorithm 3.*

Proof. Theorem 1 proceeds by applying the RDP composition theorem to individual DP-SGD steps, whose RDP is analyzed directly. We apply Lemma 2 in place of standard RDP composition. Lemma 3 shows that every time we wish to apply Lemma 2, the datasets indeed satisfy the precondition of being neighboring, with one subtlety: the indices of deleted samples (which are effectively replaced by a dummy sample with zero gradient) are not guaranteed to be coupled. However, Algorithm 1 remains unchanged in distribution if after every filtering step, we randomly permute the dataset (dummy samples included). We can thus couple the random permutations between runs of Algorithm 1 on neighboring datasets, so that the dummy indices are always aligned. This coupling does not affect the privacy proof, as it preserves marginal distributions.

□

Algorithm 3: $\mathsf{DPSGDPoisson}(\mathcal{D}, \theta_0, q, \eta, C, \sigma, T)$

Input: Dataset $\mathcal{D}$, model initialization θ_0, sample rate q, step size η, clip bound C, noise multiplier σ, step count T

Output: Final model parameters θ

$\theta \leftarrow \theta_0$
for $t \in [T]$ **do**
  $\mathcal{B} \leftarrow \{\}$
  for $(x, y) \in \mathcal{D}$ **do**
    $r \leftarrow \text{Bern}(q)$
    if $r = 1$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup (x, y)$
    end
  end
  $\xi \sim \mathcal{N}(0, (\frac{\sigma C}{|\mathcal{B}|})^2 I)$ `// Spherical Gaussian with standard deviation` $\frac{\sigma C}{|\mathcal{B}|}$
  $\theta \leftarrow \theta - \eta(\frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \text{clip}(\nabla_\theta \ell(\theta; x, y), C) + \xi)$ `//` $\texttt{clip}(v, C) = v \cdot \min(1, \frac{C}{\|v\|_2})$
end
return θ

A.2 Variants of DP-SGD

In this section, we briefly comment on the applicability of the framework in Appendix A.1 to variants of DP-SGD that see use in practice, beyond the Poisson-subsampled variant in Algorithm 3.

First, consider the variant using randomly-shuffled minibatches (as discussed in Section 2, with pseudocode in Algorithm 4). It is straightforward to check that, due to the random permutation at the start of every epoch (Line 3), the privacy framework of Appendix A.1 applies if filtering is performed after any number of gradient steps are taken. This is true whether the privacy accounting is done using the standard (formal) accounting based on RDP of the Gaussian mechanism, or the (heuristic) "in-expectation equivalent" variant of Theorem 1 discussed in Appendix A.1. In Algorithm 5, we give pseudocode of one potential implementation of a filtered variant of Algorithm 4.

Algorithm 4: $\mathsf{DPSGDShuffle}(\mathcal{D}, \theta_0, m, \eta, C, \sigma, N)$

Input: Dataset $\mathcal{D}$, model initialization θ_0, batch size m, step size η, clip bound C, noise multiplier σ, epoch count N

Output: Final model parameters θ

1 $\theta \leftarrow \theta_0$
2 **for** $t \in [N]$ **do**
3 $\{(x_i, y_i)\}_{i\in[n]} \leftarrow$ randomly permuted ordering of elements of $\mathcal{D}$
4 $\mathcal{B}_j \leftarrow \{(x_i, y_i)\}_{i\in[n]|j=\lceil \frac{i}{m} \rceil}$ for all $j \in [\lceil \frac{n}{m} \rceil]$
5 **for** $j \in [\lceil \frac{n}{m} \rceil]$ **do**
6 $\xi \sim \mathcal{N}\left(0, \left(\frac{\sigma C}{|\mathcal{B}_j|}\right)^2 I\right)$
7 $\theta \leftarrow \theta - \eta(\frac{1}{|\mathcal{B}_j|} \sum_{(x_i,y_i)\in\mathcal{B}_j} \text{clip}(\nabla_\theta \ell(\theta; x_i, y_i), C) + \xi)$
8 **end**
9 **end**
10 **return** θ

Second, we observe that unless provable privacy accounting methods change substantially, our framework is likely to extend to future (permutation-invariant) variants of DP-SGD that are developed. Indeed, any algorithm that is based on an iterative optimization method, whose individual steps are history-dependent and data-dependent, presumably requires an adaptive composition property. In essentially all existing privacy frameworks, adaptive composition is proven through an argument

Algorithm 5: FilteredDPSGDShuffle($\mathcal{D}, \theta_0, m, \eta, C, \sigma, N, f, k,$ `scope`)

Input: Dataset $\mathcal{D}$, model initialization θ_0, batch size m, step size η, clip bound C, noise multiplier σ, epoch count N, sample signature f, selection parameter k, flag `scope` $\in \{$`global`, `class`$\}$

Output: Final model parameters θ

$(\mathcal{F}, S, \theta) \leftarrow (\emptyset, \{0\}^{|\mathcal{D}|}, \theta_0)$

for $t \in [N]$ **do**

  $\{(x_i, y_i)\}_{i \in [n]} \leftarrow$ randomly permuted ordering of elements of $\mathcal{D}$

  $\mathcal{B}_j \leftarrow \{(x_i, y_i)\}_{i \in [n] | j = \lceil \frac{i}{m} \rceil}$ for all $j \in [\lceil \frac{n}{m} \rceil]$

  for $j \in [\lceil \frac{n}{m} \rceil]$ **do**

    foreach $(x_i, y_i) \in \mathcal{B}_j$ **do**

      $g_i \leftarrow \begin{cases} \text{clip}(\nabla_\theta \ell(\theta; x_i, y_i), C) & i \notin \mathcal{F} \\ 0 & i \in \mathcal{F} \end{cases}$

      $S[i] \leftarrow f_\theta(x_i, y_i)$

    end

    $\xi \sim \mathcal{N}(0, (\frac{\sigma C}{|\mathcal{B}_j|})^2 I)$

    $\theta \leftarrow \theta - \eta(\frac{1}{|\mathcal{B}_j|} \sum_{(x_i, y_i) \in \mathcal{B}_j} g_i + \xi)$

  end

  $\mathcal{F} \leftarrow \mathcal{F} \cup \mathsf{SelectTop}(S, \mathcal{D}, \mathcal{F}, k,$ `scope`$)$

end

return θ

compatible with Lemma 2: after conditioning on the outcome of the first mechanism, the proof no longer requires that the datasets be the same in the second mechanism.

To give an example of this phenomenon, consider the recent algorithm of [CGH+25]. This scheme is also permutation-invariant, and its privacy (as a *random allocation scheme*) is established using RDP and PLD [FS25, FS26]. These accounting frameworks (particularly, their proofs of adaptive composition) are compatible with Lemma 2, and thus a variant of [CGH+25] using sample signature-based filtering would inherit the privacy analyses of [FS25, FS26].

B Empirical Privacy Reporting Details

In this section, after describing our experimental setup explicitly in Appendix B.1, we give a detailed overview of our methodology used to elicit estimates ε_{lb} on algorithms' empirical privacy.

In Appendix B.2, we provide a general overview of f-DP and its use in evaluating LiRA-style membership inference attacks. [CCN+22, NHS+23]. In Appendix B.3, we explain common heuristics used when reporting ε_{lb} values in the privacy auditing literature, and caution against their potential pitfalls. Our paper always reports ε_{lb} estimates both with and without these heuristics, for all algorithms. Finally, in Appendix B.4, we explain some numerical precision oddities that arise when applying Clopper-Pearson corrections to finite sample estimation procedures.

B.1 Experimental and audit setup

In this section, we describe the experimental setup used to evaluate our defense. Then, we evaluate our defense framework against a suite of privacy audits.

Datasets. We evaluate audits and defenses on three standard supervised learning benchmarks spanning both vision and tabular domains. For image classification, we use MNIST [LBBH98], a grayscale handwritten digit dataset with 10 classes and 28×28 inputs, and CIFAR-10 [Kri09], a natural image dataset with 10 classes and 32×32 RGB inputs. We also use Purchase100 [SSSS17], a tabular dataset of customer purchase histories. Each example is represented as a binary feature vector indicating the presence/absence of purchased items, with 100 class labels. Licenses for these datasets are not entirely clear, with some inconsistent claims depending on source, though all are considered acceptable for non-commercial academic use, as in this paper.

Initializations. We reuse the same model initialization across all training runs for each experiment. Fixed initialization reduces run-to-run variability, which makes it easier to attribute differences in the final model to the data (e.g., whether a canary was included) rather than to randomness in initialization. We use Xavier initialization [GB10] for CNN-based models and Kaiming initialization [HZRS15] for WideResNet models [ZK16]. All models are trained from scratch.

Model architectures. For MNIST, we use a convolutional neural network (CNN). For CIFAR-10, we test both a CNN and a WideResNet architecture to demonstrate robustness across different model families. For Purchase100, we use a 3-layer multilayer perceptron (MLP).

Training configuration. For all experiments, unless otherwise specified, we use privacy parameters $\varepsilon = 10$ and $\delta = 10^{-5}$ for DP-SGD and our composed defense. We train for $N = 100$ epochs using shuffled mini-batches. We perform hyperparameter grid search to identify learning rates that maximize test accuracy: $\eta = 3$ for MNIST and CIFAR-10, and $\eta = 10$ for Purchase100. To disentangle improvements due to the proposed defense from improvements due to tuning, we use the same hyperparameter configurations when auditing with and without our defense.

Defense configuration. For all experiments, we run our defense using the L_∞ norm as the scoring function (Algorithm 5, Line 8) unless otherwise specified. Ablation studies (Appendix F) show that alternative scoring functions perform comparably. We use per-class (local) filtering rather than global filtering to align with prior work [TLM18]. We find that for blank canaries, global filtering also works if we change the scoring function (Appendix K.1). For MNIST and CIFAR10, we filter 5 samples from each class per epoch, amounting to dropping 8.3% and 10% of each training set, respectively. For Purchase100, we filter 1 sample from each class per epoch, amounting to dropping 6.5% of the training set. The choice of #samples to drop per class per epoch was arbitrary, as long as the total #samples dropped was reasonable ($\leq 10\%$). We find that for blank canaries, our defense is robust to filtering frequency (Appendix K.3) and "bandwidth" (# samples discarded, Appendix K.2).

Augmentation multiplicity. Practitioners sometimes generate multiple augmented versions of the same sample within a single batch and average their gradients before clipping, a technique known as augmentation multiplicity. This approach is used to improve the privacy-utility tradeoff by reducing the variance of the gradient estimate for each sample. While we do not employ augmentation multiplicity in all of our experiments, our evaluation of our defense includes this setting (cf. Table 1) to ensure it remains effective against this commonly-used privacy-enhancing modification.

Audit models and evaluation. We train $k = 400$ models for each audit: 200 with the canary and 200 without. While some prior works train more models, we find that with the heuristics in Appendix B applied, 400 shadow models offers a sufficient scale to observe a substantial gap in attack success between filtered and unfiltered settings. Our alternative ε_{lb} reporting in Appendix L shows that even without these heuristics, a smaller but noticeable gap persists. As validation, we also provide an MNIST audit with 1000 models (Appendix J), which shows limited improvement in audit tightness. Our primary focus is the relative gap in empirical privacy with versus without the filter, so we adopt the same methodology for auditing all optimizers.

Compute resources. All experiments were run on H200 GPUs. Individual audit jobs used between 5 and 20 GPUs and ran for 5–30 hours, with resource requirements scaling according to model architecture and dataset size. Specific job configurations (number of GPUs, hours per job, batch scripts) are available in the scripts/directory of our code repository.

B.2 Measuring empirical privacy

We review LiRA-style membership inference attacks for DP auditing [CCN$^+$22, NHS$^+$23].

Trade-off functions. Consider two arbitrary distributions P, Q supported on the same space Ω, and a *decision rule* $\phi : \Omega \to [0, 1]$. Intuitively, ϕ acts as a distinguisher between P and Q, where $\phi(\omega) = 0$ can be taken as guessing that ω was sampled from P, and $\phi(\omega) = 1$ indicates a sample from Q (fractional values thus represent uncertainty). We accordingly define

$$\alpha_{P,Q}(\phi) := \mathbb{E}_{\omega\sim P}[\phi(\omega)], \quad \beta_{P,Q}(\phi) := \mathbb{E}_{\omega\sim Q}[1 - \phi(\omega)]$$

to be the type I (FPR) and type II (FNR) error rates of ϕ. For each FPR $\alpha \in [0, 1]$, we can consider the optimal test which minimizes the FNR $\beta \in [0, 1]$ at the specified α level, formalized as follows.

Definition 3 (Trade-off function)**.** *For two distributions P, Q supported on Ω, we define the* trade-off function $T_{P,Q} : [0, 1] \to [0, 1]$ *by* $T_{P,Q}(\alpha) := \inf_{\phi:\Omega\to[0,1],\alpha_{P,Q}(\phi)\leq\alpha} \beta_{P,Q}(\phi)$.

More generally, we say that $f : [0,1] \to [0,1]$ is a trade-off function if $f = T_{P,Q}$ for some P, Q.

The Neyman-Pearson lemma characterizes ϕ achieving the infimum in $T_{P,Q}$ as thresholded likelihood ratio tests (Theorem A.1, [DRS22]). Next, we state some basic properties of trade-off functions.

Lemma 4 (Proposition 2.2, [DRS22])**.** *$f : [0,1] \to [0,1]$ is a trade-off function iff it is convex, non-increasing, and satisfies $f(\alpha) \leq 1 - \alpha$ for all $\alpha \in [0,1]$.*

The first property in Lemma 4 follows because given tests ϕ, ϕ' with FPRs α, α', and mixing parameter $\lambda \in [0,1]$, we can always define a test that outputs ϕ with probability λ and ϕ' with probability $1 - \lambda$. The second property is because the constraint set on ϕ is only larger as α grows.

Finally, the last property is because we can always simultaneously obtain FPR α and FNR $1 - \alpha$, for any $\alpha \in [0,1]$, by ignoring the sample and uniformly setting $\phi = \alpha$. If $f(\alpha) = 1 - \alpha$ uniformly, then P and Q are undistinguishable by any test, i.e., $P = Q$ almost everywhere.

f-DP and (ε, δ)-DP. The definition of f-DP is naturally motivated by Definitions 1 and 3.

Definition 4 (f-DP)**.** *Let $f : [0,1] \to [0,1]$ be a trade-off function. We say that a mechanism $\mathcal{M} : \mathcal{S}^n \to \Omega$ satisfies f-differential privacy (or, is f-DP) if for all neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$,*

$$T_{\mathcal{A}(\mathcal{D}),\mathcal{A}(\mathcal{D}')}(\alpha) \geq f(\alpha) \text{ for all } \alpha \in [0,1]. \tag{5}$$

A higher trade-off function corresponds to distributions being difficult to distinguish. Definition 4 captures this intuition by positing that every pair $\mathcal{A}(\mathcal{D})$, $\mathcal{A}(\mathcal{D}')$ induced by neighboring datasets is at least hard to distinguish as specified by f. For example, if $f(\alpha) = 1 - \alpha$ uniformly, then any f-DP $\mathcal{A}$ has $\mathcal{A}(\mathcal{D}) = \mathcal{A}(\mathcal{D}')$ almost everywhere, so its outcome is independent from the dataset it is trained on with probability 1. We call $\mathcal{A}$ that satisfies such an f-DP *perfectly private*.

Importantly, [WZ10, DRS22] observed the following equivalence between Definitions 1 and 4.

Lemma 5 (Proposition 2.12, [DRS22])**.** *Let $f : [0,1] \to [0,1]$ be a symmetric trade-off function. Then a randomized algorithm $\mathcal{A}$ is f-DP iff for all $\delta \in (0,1)$, $\mathcal{A}$ is $(\varepsilon(\delta), \delta)$-DP, where*

$$\varepsilon(\delta) := \max\left\{0, \max_{\alpha \in [0,1]} \log\left(\frac{1 - f(\alpha) - \delta}{\alpha}\right)\right\}, \tag{6}$$

Note that if we increase f pointwise, applying (6) leads to smaller privacy parameters.

Pipeline. Following [NST+21, NHS+23], we can certify privacy lower bounds via Lemma 5. Recall that the true DP function $\varepsilon^\star$ (1) of an algorithm $\mathcal{A}$ are the largest parameters attainable via a set of tests (corresponding to a dataset-canary-audit triple (2) at each $\delta \in (0,1)$). Analogously, one can define the true f-DP curve to be the pointwise-smallest f so that $\mathcal{A}$ is f-DP. Recalling (5), each point on this true curve is again realized by a dataset, canary, and audit $\phi : \Omega \to [0,1]$.

In practice we can only realize a subset of all possible dataset-canary-audit triples, and thus the best we can do is to upper bound the true f-DP curve as tightly as possible. Each test is parameterized by neighboring datasets $(\mathcal{D}, \mathcal{D}')$ and a candidate test $\phi : \Omega \to [0,1]$ where Ω is the output space of $\mathcal{A}$. Suppose we ascertain that for this particular triple,

$$\mathbb{E}_{\omega \sim \mathcal{A}(\mathcal{D})}[\phi(\omega)] \leq \hat{\alpha}, \quad \mathbb{E}_{\omega \sim \mathcal{A}(\mathcal{D}')}[1 - \phi(\omega)] \leq \hat{\beta}.$$

A standard way to obtain $(\hat{\alpha}, \hat{\beta})$ is to generate holdout copies of $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$, estimate the two expectations above from empirical averages, and apply a Clopper-Pearson correction. This certifies $f^\star(\alpha) \leq \hat{\beta}$ for all $\alpha \geq \hat{\alpha}$, where $f^\star$ denotes the true f-DP curve for $\mathcal{A}$. Any such triple $(\phi, \mathcal{D}, \mathcal{D}')$ yields an empirical upper bound on $f^\star$, which translates to a lower bound on $\varepsilon^\star$ via (6).

A well-documented difficulty in carrying out this methodology is that unless the number of holdout copies used is (very) large, the Clopper-Pearson correction can dramatically loosen the audit. As a result, prior works have adopted several heuristic approximations (multiple hypothesis testing, and use of Gaussian DP curves) that can inflate the estimated ε_{lb}, while sometimes sacrificing formal correctness. We discuss these heuristics at length next, in Appendix B.3, and provide counterexamples flagging situations where their guarantees do not formally apply.

We make explicit that in the rest of the paper (up until Appendix L), all of our reported ε_{lb} use *both* the multiple hypothesis testing and GDP extrapolation heuristics described in Appendix B.3. This

reporting is consistent with prior privacy auditing works advertising tight or nearly-tight audits, which also used these heuristics [NHS+23, AC24, CBP25]. Notably, by using both heuristics, the reported ε_{lb} metrics of our algorithm (as well as all other defenses) is as unfavorable as possible.

For completeness, in Appendix L we report *all* of the estimated ε_{lb} for DP-SGD and our filtered variants, both with and without these heuristics, and using different sample splits. Notably, our filtered DP-SGD variant consistently achieves a lower ε_{lb} than the unfiltered implementation against input space, hidden state audits across all reporting methods (formal or not).

B.3 Heuristic conventions and their (potential) pitfalls

In this section, we detail two practices often used in the privacy auditing literature, and point out when their guarantees formally apply (contrasted with when they should be treated as a heuristic). While this discussion is sometimes present in prior works, we believe making it explicit, with specific counterexamples to the validity of heuristics, will prove valuable to future auditing research, as well as practitioners reasoning about the guarantees of these reporting metrics.

Following Appendix B.2, the formal way to certify a privacy lower bound on an algorithm $\mathcal{A}$ is as follows. First, a test ϕ however the auditor desires (e.g., the most effective loss threshold is selected using a set of models, which are discarded). Next, fresh holdout models are trained using $\mathcal{A}$ and evaluated according to the test ϕ, producing an empirical FPR-FNR tradeoff (α, β). Then, a Clopper-Pearson correction with failure probability γ is applied to obtain $(\hat{\alpha}, \hat{\beta})$ lying above the f-DP curve with probability $1 - \gamma$. Finally, (6) converts this point into an ε_{lb} estimate. The two main heuristics we discuss in this section are *multiple hypothesis testing* and *GDP extrapolation*. We describe these two heuristics in turn, and then examine their potential pitfalls.

Multiple hypothesis testing. Algorithm 6 provides pseudocode for the *multiple hypothesis testing* heuristic, where the same models are used for selecting the test ϕ, and evaluating it to produce a point $(\hat{\alpha}, \hat{\beta})$ above the f-DP curve. Observe that the same k i.i.d. copies are used in all of the tests $\{\phi_i\}_{i\in[m]}$, and then the best ε_{lb} estimate amongst these tests is reported using (6). It is formally incorrect to say that the selected pair $(\hat{\alpha}_i, \hat{\beta}_i)$ lies above the true f-DP curve with probability $1 - \gamma$, due to the potential for false discovery. The formal fix is to either use a Bonferroni correction (union bound) and set the coverage of each Clopper-Pearson correction in Algorithm 6 to $1 - \frac{\gamma}{m}$, or to apply a selected test ϕ_i to holdout samples (with Clopper-Pearson coverage $1 - \gamma$). Indeed, the original work of [JUO20] prescribes using the holdout strategy to ensure statistical validity.

Algorithm 6: $\mathsf{EmpiricalDPLB}(\mathcal{A}, \{\mathcal{D}_i, \mathcal{D}'_i, \phi_i\}_{i\in[m]}, \delta, \gamma, k)$

Input: Randomized algorithm $\mathcal{A} : \mathcal{X}^n \to \Omega$, neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^*$ and m tests $\phi_i : \Omega \to [0, 1]$, $\delta \in (0, 1)$, $\gamma \in (0, 1)$, $k \in \mathbb{N}$
Output: Empirical privacy lower bound $\varepsilon_{\text{lb}}(\delta)$
$\{\omega_j\}_{j\in[k]} \sim_{\text{i.i.d.}} \mathcal{A}(\mathcal{D})$
$\{\omega'_j\}_{j\in[k]} \sim_{\text{i.i.d.}} \mathcal{A}(\mathcal{D}')$
for $i \in [m]$ **do**
  $\alpha_i^{\text{init}} \leftarrow \frac{1}{k}\sum_{j\in[k]} \phi_i(\omega_j)$
  $\beta_i^{\text{init}} \leftarrow 1 - \frac{1}{k}\sum_{j\in[k]} \phi_i(\omega'_j)$
  $(\hat{\alpha}_i, \hat{\beta}_i) \leftarrow (\text{Beta}^{-1}_{k\alpha_i^{\text{init}}+1, k(1-\alpha_i^{\text{init}})}(1-\gamma), \text{Beta}^{-1}_{k\beta_i^{\text{init}}+1, k(1-\beta_i^{\text{init}})}(1-\gamma))$
  `// Clopper-Pearson, Beta`$_{a,b}$ `is CDF of beta distribution with parameters` a, b
end
Return: $\varepsilon_{\text{lb}}(\delta) \leftarrow \max_{i:\hat{\alpha}_i>0} \log(\frac{1-\hat{\beta}_i-\delta}{\hat{\alpha}_i})$

Subsequent works in privacy auditing are less careful on this matter, with many (e.g., [MSS23, ZBWT+23, NHS+23, AC24, CBP25]) utilizing a variant of the multiple hypothesis testing method in Algorithm 6, *without* explicitly using a Bonferroni correction or holdouts. Indeed, Section 5.2 of [NHS+23] explicitly flags this issue, where they state that selecting an audit in this way is "technically not valid," but that "it has become common" to report lower bounds in this way.

Our experiments (cf. Appendix L) suggest that the Bonferroni correction yields substantially smaller (often negligible) ε_{lb} estimates. On the other hand, the holdout method still results in nontrivial formal ε_{lb} estimates when auditing models trained with DP-SGD, although the tightness of these estimates often worsens by a factor of 3-10×. We thus caution against taking reported ε_{lb} wholly at face value if multiple discovery is not accounted for, at least without ablations justifying the heuristic. For completeness, we report estimated ε_{lb} both with and without multiple discovery, as trends between our algorithms and DP-SGD remain consistent across reporting methods.

Another subtlety regarding multiple discovery that arises in the practice of privacy auditing (and private ML more broadly) is *hyperparameter selection*, see e.g., [MSH+22, PS22, PHK+23]. To circumvent this issue, all of our hyperparameter selection is explicitly performed using holdout models that are discarded before selecting our audits and reporting our ε_{lb} estimates.

Finally, we make one observation that is implicit in prior works using Algorithm 6, but which has not received explicit discussion to our knowledge. A reasonable modification to Line 10 of Algorithm 6 uses Lemma 4, i.e., it first computes the largest convex function f below the points $\{(\hat{\alpha}_i, \beta_i)\}_{i\in[m]}$, before applying the formula (6). This modification could in principle yield tighter audits, as it leverages the ability to use points on the f-DP curve outside the $\{(\hat{\alpha}_i, \hat{\beta}_i)\}_{i\in[m]}$. As shown in Lemma 6, this convexification step in fact does not change the computed ε_{lb}.

Lemma 6. *Let $\mathcal{S} = \{(\alpha_i, \beta_i)\}_{i\in[n]} \subset [0,1]^2$ have distinct first coordinates, and let f be the lower convex envelope of $\mathcal{S}$ (the pointwise largest convex function with $f(\alpha_i) \le \beta_i$ for all $i \in [n]$).*[6] *Also, assume that $\{(0,1),(1,0)\} \subset \mathcal{S}$. Then for all $\delta \in (0,1)$,*

$$\sup_{x\in(0,1]} \log\left(\frac{1-f(x)-\delta}{x}\right) = \max_{i:\alpha_i>0} \log\left(\frac{1-\beta_i-\delta}{\alpha_i}\right),$$

with the convention that the logarithm of a nonpositive number is $-\infty$.

Proof. Throughout the proof, fix some $\delta \in (0,1)$ and define

$$\psi(x) := \log\left(\frac{1-f(x)-\delta}{x}\right), \quad M_1 := \sup_{x\in(0,1]} \psi(x), \quad M_2 := \max_{i:\alpha_i>0} \log\left(\frac{1-\beta_i-\delta}{\alpha_i}\right).$$

It is immediate that $M_2 \le M_1$, because $\beta_i \ge f(\alpha_i)$ and α_i is a candidate argument in the definition of M_1. For the reverse inequality, the lower convex envelope of finitely many points is piecewise affine, and its breakpoints occur at points α_i where the constraint is tight, i.e., $f(\alpha_i) = \beta_i$ (see Theorem 19.1, [Roc97]). Therefore, it is enough to show that on an interval $x \in [\alpha_i, \alpha_j]$ such that $f(x) = ax + b$, $f(\alpha_i) = \beta_i$, and $f(\alpha_j) = \beta_j$, ψ achieves its maximum at an endpoint α_i or α_j.

To see this last fact, observe that on this interval,

$$\psi(x) = \log\left(\frac{1-b-\delta-ax}{x}\right), \quad \psi'(x) = -\frac{1-b-\delta}{x(1-b-\delta-ax)}, \quad \text{for } x < \frac{1-b-\delta}{a}.$$

Hence ψ' has a fixed sign on this interval (whenever $\psi > -\infty$), so ψ is supremized at an endpoint. □

Gaussian DP (GDP) extrapolation. GDP curves [DRS22] are a family of f-DP curves that are particularly analytically convenient. Informally, these are a one-parameter family of f-DP curves that exactly capture the trade-off functions $T_{P,Q}$ (Definition 3) when $P = \mathcal{N}(0,1)$ and $Q = \mathcal{N}(\mu,1)$ are equal-variance Gaussians; here, the one parameter is μ. This family is relevant to our setting because DP-SGD utilizes the Gaussian mechanism, and the composition of GDP curves remains a GDP curve (Corollary 3.3, [DRS22]). GDP curves are notably compatible with sample-efficient auditing: if the true privacy loss curve is GDP, then the auditor can extrapolate from a single tradeoff point (α, β) to rule out part of the GDP parametric family, improving statistical efficiency. Consequently, prior works starting from [NHS+23] (see also [AC24, CBP25]) propose using GDP methodology to achieve tighter audits. We call this strategy *GDP extrapolation*, and note that it should be treated as a heuristic whenever the true f-DP curve is not a GDP curve.

In this section, we point out two potential pitfalls to using this methodology to formally audit DP-SGD variants. These pitfalls stem from the facts that GDP curves are *not* closed under the following

[6] The lower convex envelope is uniquely defined, see Exercise 3.30 of [BV04].

operations: (1) Poisson subsampling, and (2) marginalizing intermediate outputs (i.e., in the hidden state threat model). We formalize these points with explicit counterexamples in Corollaries 2 and 3. Prior works have also discussed these issues to some extent, e.g., Section 5.5 of [NHS+23], and an extended discussion of the subsampling issue in [GKK+26].

One common justification for use of GDP-based auditing, when the f-DP curve is not formally a GDP curve, is the f-DP central limit theorem (Theorem 3.4, [DRS22]), as essentially all private optimizers in ML are iterative methods that successively apply the same mechanism. We caution that again, the proof of the f-DP central limit theorem relies on accessing intermediate outputs (which becomes problematic under hidden state threats, e.g., as in [CBP25]), and hence this logic is susceptible to the counterexample in Corollary 3. Altogether, GDP-based auditing requires careful justification depending on the setup; for this reason, our experiments report both the GDP-based ε_{lb} and a weaker, but more formal, estimate based on Clopper-Pearson correction.

Let Φ denote the CDF of a standard Gaussian, and let $\varphi = \Phi'$ denote its PDF. The following lemma is the basis of our counterexamples, and states that the trade-off function between a particular Gaussian and mixture of two Gaussians is not a GDP curve.

Lemma 7. *Consider the trade-off function $T_{P,Q}(\alpha)$ for $P = \mathcal{N}(0,1)$ and $Q = \frac{1}{2}\mathcal{N}(0,1) + \frac{1}{2}\mathcal{N}(1,1)$. There does not exist a $\mu > 0$ such that $T_{P,Q} = G_\mu$, where $G_\mu(\alpha) := \Phi(\Phi^{-1}(1-\alpha) - \mu)$.*

Proof. By the Neyman–Pearson lemma, optimal tests threshold the likelihood ratio. Here

$$\frac{dQ}{dP}(x) = \frac{1}{2} + \frac{1}{2}\exp\left(x - \frac{1}{2}\right),$$

which is strictly increasing in x. Therefore, for false-positive rate $\alpha = \Pr[X \geq t] = 1 - \Phi(t)$, the optimal test is $\mathbb{I}_{x \geq t}$, , where $\mathbb{I}_{\mathcal{E}}$ is the indicator of an event $\mathcal{E}$, and its false-negative rate is

$$T_{P,Q}(\alpha) = \Pr_{X \sim Q}[X < t] = \frac{1}{2}\Phi(t) + \frac{1}{2}\Phi(t-1), \qquad t = \Phi^{-1}(1-\alpha).$$

If $T_{P,Q} = G_\mu$, then for all $t \in \mathbb{R}$, $\frac{1}{2}\Phi(t) + \frac{1}{2}\Phi(t-1) = \Phi(t-\mu)$. Differentiating gives

$$\frac{1}{2}\varphi(t) + \frac{1}{2}\varphi(t-1) = \varphi(t-\mu).$$

Dividing by $\varphi(t)$,

$$\frac{1}{2} + \frac{1}{2}e^{t-\frac{1}{2}} = e^{t\mu - \frac{\mu^2}{2}}.$$

As $t \to -\infty$, the two sides tend to $\frac{1}{2}$ and 0 respectively for every $\mu > 0$, a contradiction. □

Our claimed counterexamples now follow simply from Lemma 7.

Corollary 2. *The trade-off function for the Gaussian mechanism with Poisson subsampling does not always correspond to a μ-GDP trade-off function for some $\mu > 0$.*

Proof. Consider a single-element dataset, either 0 or 1, and Poisson subsampling with parameter $q = \frac{1}{2}$ composed with the Gaussian mechanism (adding $\mathcal{N}(0,1)$ to the empirical mean). If the dataset is 0, then the resulting distribution is $P = \mathcal{N}(0,1)$, while if the dataset is 1, then the resulting distribution is $Q = \frac{1}{2}\mathcal{N}(0,1) + \frac{1}{2}\mathcal{N}(1,1)$, which is exactly the example described in Lemma 7. □

Corollary 3. *Consider composing several Gaussian mechanisms (adaptively) and outputting only the result of the final mechanism. The trade-off function of this mechanism does not always correspond to a μ-GDP trade-off function for some $\mu > 0$.*

Proof. Consider a two-stage Gaussian mechanism, with a single-element dataset d that is either 0 or 1. The first stage is to draw $X \sim \mathcal{N}(0,1)$, independent of the dataset. We adaptively choose the second stage based on the result of the first: we output a sample from $\mathcal{N}(d \cdot \mathbb{I}_{\{X \geq 0\}}, 1)$. Considering the various choices of d, this again results in exactly the aforementioned distributions $P = \mathcal{N}(0,1)$ or $Q = \frac{1}{2}\mathcal{N}(0,1) + \frac{1}{2}\mathcal{N}(1,1)$. □

B.4 Numerical precision

Applying a fully formal empirical privacy estimation methodology, e.g., Clopper–Pearson corrections after holdout, can sometimes report highly-conservative ε_{lb} figures that sometimes even show up as vanishing. We numerically examine how the Clopper–Pearson (CP) correction changes empirical privacy lower bounds in the finite-sample regime. For this discussion, we fix the significant level of confidence internal to $\gamma = 0.05$ and ignore the union bound over multiple tests (i.e., the false discovery direction in Appendix B.3, so that each of the two error rates, α and β, is corrected using a one-sided CP upper confidence bound with failure probability $\frac{\gamma}{2} = 0.025$. For each sample size $k \in \{200, 500, 1000, 5000, 10000\}$, we enumerate all possible empirical count pairs $(x_\alpha, x_\beta) \in [k] \times [k]$, where $\alpha_{\text{emp}} = \frac{x_\alpha}{k}, \beta_{\text{emp}} = \frac{x_\beta}{k}$. For each pair, we compute the raw empirical lower bound and the CP corrected lower bound:

$$\varepsilon_{\text{raw}} = \max\left\{0, \log\frac{1-\beta_{\text{emp}}-\delta}{\alpha_{\text{emp}}}\right\}, \quad \varepsilon_{\text{CP}} = \max\left\{0, \log\frac{1-\hat{\beta}-\delta}{\hat{\alpha}}\right\},$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the CP upper confidence bounds, and restrict attention to count pairs with $\varepsilon_{\text{raw}} > 0$. This isolates empirical observations that initially provide a nonzero privacy signal and shows how many of them are weakened or collapsed to zero after CP correction. We show our result in Figure 2. We make two observations from the numerical results.

1. In the experiment, we filter out all (α, β) pairs for which $\varepsilon_{\text{raw}} = 0$ or $\varepsilon_{\text{raw}} = +\infty$. After this filtering, the range of finite empirical lower bounds that can appear is still limited by the sample size. In particular, the largest possible auditable ε_{lb} grows only logarithmically with k: it is around 4 for $k = 200$ in our plotted range and around 9 for $k = 10000$.
2. As k increases, fewer nonzero raw observations collapse to zero after CP correction, and the average gap between ε_{raw} and ε_{cp} also decreases. However, the worst-case raw value among the collapsed observations does not vanish; in our experiments, it remains around 2.4.

Both observations have simple theoretical explanations. For the first observation, over all finite count pairs with sample size k, the largest possible raw lower bound is attained when $x_\alpha = 1$ and $x_\beta = 0$, giving

$$\varepsilon^\star_{\text{raw}} = \log\frac{1-\delta}{1/k} = \log\big((1-\delta)k\big).$$

Thus, the auditable range of finite empirical lower bounds grows only logarithmically with the number of samples, suggesting that accurately auditing large ε values requires very large k.

For the second observation, the non-vanishing worst-case collapse is caused by extreme low-count configurations near the boundary of the binomial experiment. Let $x_\alpha = 1, x_\beta = k - c$, where c is some constant that will be specified later. Equivalently $\alpha_{\text{emp}} = \frac{1}{k}, 1 - \beta_{\text{emp}} = \frac{c}{k}$. Therefore the raw lower bound is $\varepsilon_{\text{raw}} = \log\frac{c/k-\delta}{1/k}$. When δ is negligible compared with c/k, this is approximately $\varepsilon_{\text{raw}} \approx \log c$, which is independent of k. We then consider the CP correction:

$$\hat{\alpha} \approx \frac{q_{1-\gamma/2}(\Gamma(2,1))}{k}, \quad \hat{\beta} \approx 1 - \frac{q_{\gamma/2}(\Gamma(c,1))}{k}.$$

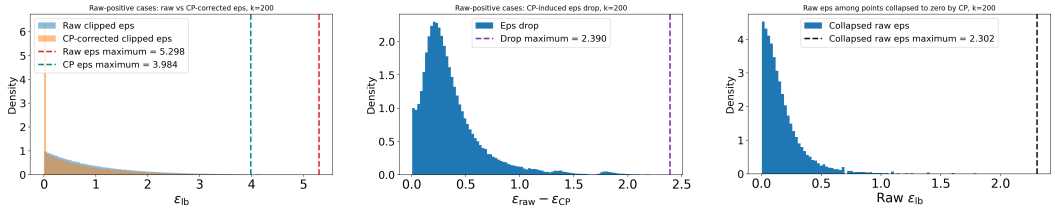
where $q_p(\Gamma(a,1))$ denotes the p-quantile of a Gamma distribution. The CP-corrected lower bound collapses to zero when $\hat{\alpha} + \hat{\beta} + \delta \geq 1$. Substituting the approximations above, the collapse condition becomes

$$q_{\gamma/2}(\Gamma(c,1)) - k\delta \lesssim q_{1-\gamma/2}(\Gamma(2,1)).$$

For $\gamma = 0.05$, the right-hand side is approximately 5.57. When $k\delta$ is small, the largest integer c satisfying this condition is about $c = 11$, giving

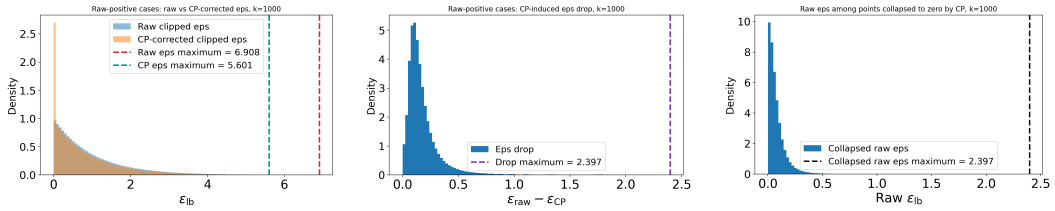
$$\varepsilon_{\text{raw}} = \log(c - k\delta) \approx \log 11 \approx 2.4.$$

Thus, the non-vanishing maximum collapsed value is a finite-count boundary effect: it persists for large k as long as $k\delta$ remains small relative to the relevant true-positive count c. In the pure-DP case ($\delta = 0$), this gap persists for any choice of finite k. In the common auditing setting with $\delta = 10^{-5}$,



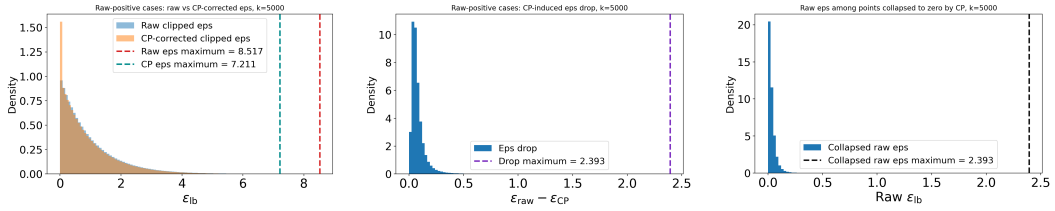
ε_{raw} v.s. ε_{cp} $\varepsilon_{\text{raw}} - \varepsilon_{\text{CP}}$ ε_{raw} among collapse points

(a) $k = 200$



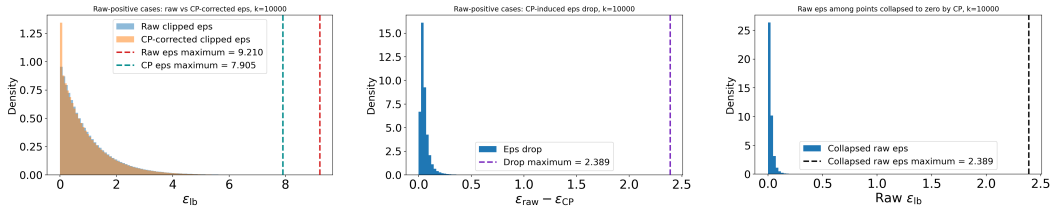
ε_{raw} v.s. ε_{cp} $\varepsilon_{\text{raw}} - \varepsilon_{\text{CP}}$ ε_{raw} among collapse points

(b) $k = 1000$



ε_{raw} v.s. ε_{cp} $\varepsilon_{\text{raw}} - \varepsilon_{\text{CP}}$ ε_{raw} among collapse points

(c) $k = 5000$



ε_{raw} v.s. ε_{cp} $\varepsilon_{\text{raw}} - \varepsilon_{\text{CP}}$ ε_{raw} among collapse points

(d) $k = 10000$

Figure 2: Finite-sample effect of Clopper–Pearson (CP) correction under different sample sizes k, with $\gamma = 0.05$ and $\delta = 10^{-5}$. Each row corresponds to a fixed k. The left column compares the distributions of the raw lower bound ε_{raw} and the CP-corrected lower bound ε_{cp}; the middle column shows the distribution of the correction gap $\varepsilon_{\text{raw}} - \varepsilon_{\text{cp}}$; and the right column shows the distribution of raw lower bounds that collapse to 0 after CP correction.

the gap begins to disappear only when $k\delta$ becomes comparable to c; for $c \approx 11$, this requires $k \gtrsim 10^6$, which is a rather demanding sample size.

Our remaining appendices provide additional experimental results and ablation studies that complement our main findings. We summarize the different appendices here.

- **C: Gradient Space Canary.** We move to a substantially stronger *gradient space* threat model, and critically examine potential weaknesses of our framework in this setting.

- **D: Disparate Impact.** Exploring how defense targets minority vs majority subgroups for sample removal, for varying minority/majority ratios.
- **E: Alternative Input Space Audits.** Evaluations of input space audit constructions in the multiple canary setting, and the randomized canary setting of [SNJ23].
- **F: Sample Signatures.** Ablation study comparing different scoring functions for filtering.
- **G: Poisson vs. Shuffled Subsampling.** Comparison of DP-SGD subsampling variants, and the effect of DP-SGD implementation changes on our estimated ε_{lb}.
- **H: Interpolating Canaries.** Testing defense effectiveness across different scalings of canary strengths on a mixture of a blank canary and a natural image.
- **I: Auditing via Non-Canary Samples.** Analyzing privacy loss on non-canary samples, motivated by the "privacy onion effect" observed by [CJZ+22].
- **J: Increasing Trial Count.** Validating audit methodology with 1000 shadow models.
- **K: Additional Hyperparameter Tuning.** Ablations on global filtering, local filtering bandwidth, and filter frequency.
- **L: Empirical Privacy Results Across Audit Methods.** Fully specifying ε_{lb} computed under GDP and CP methodology with different holdout ratios, as described in Appendix B.

C Gradient Space Attacks

In this section, we consider defense-aware canaries that operate under a gradient space threat model. Our goal is to qualitatively examine the effectiveness of our filtered defense when exposed to a stronger, not always realizable, threat model for canary construction. All gradient space attacks are evaluated with multi-canary audits.

Multi-canary audits. For attacks involving multiple canaries, we compute each model's score as the maximum loss over all canary samples. Note that the ε_{lb} value reported for these attacks is a measure of k-group privacy, where k denotes the number of canaries; while it is reasonable to expect the k-group privacy parameter to scale as $\approx k\times$ the 1-group privacy parameter, one cannot formally convert k-group privacy into 1-group privacy in this way.

Gradient space attacks. For gradient space attacks, we follow the construction of [CBP25], where the attacker uses a holdout model to construct canary gradients as one-hot vectors in the direction of the largest parameter update: $\arg\max_i(\Delta\theta_T - \Delta\theta_0)$, where θ_T is the final model and θ_0 is the initialization. The auditor then measures the parameter difference $\Delta\theta = \theta_T - \theta_0$ at index i, corresponding to the one-hot index of the canary gradient vector.

Gradient bandwidth attack. In this attack, we allow the attacker to observe intermediate gradient distributions and defense hyperparameters over the course of training for some holdout model. The attacker first tracks s_t, the 6th largest gradient norm within class 0 at each epoch t. Then, the attacker constructs 500 copies of a one-hot gradient space canary, each with L_∞ norm equal to $\min_t s_t$ (the minimum 6th largest gradient norm observed across all training epochs). This construction exemplifies an attacker with the specific goal of dodging our filtering bandwidth in the gradient space, full state threat model. If we choose to filter 5 samples from each class on every epoch, the gradient space canaries will remain just below the filtering threshold throughout training. Under this attack, we obtained (group privacy) $\varepsilon_{\text{lb}} = 27.26$ both with and without the defense. We emphasize that this ε_{lb} is a *group privacy* statistic, so it does not contradict the ε_{ub} setting.

Gradient canceling attack. We were able to break our defense more dramatically by using a more complex *gradient canceling* attack, designed with knowledge of our defense's hyperparameter choices. This attack exploits our filter by constructing two groups of gradient space canaries with norms that sum to zero. Group A contains 500 canaries with L_∞ norm 1, while Group B contains 600 canaries with L_∞ norm $\frac{5}{6}$ in the opposite direction. Without our defense, the two groups cancel each other out during training, yielding an ε_{lb} near zero. However, when the defense is active and the defense hyperparameters are chosen so that we only filter Group A, it inadvertently exposes Group B, resulting in non-zero privacy loss. We evaluate privacy using the group ε_{lb} (Table 4).

This attack assumes a stronger adversary with knowledge of the defense hyperparameters to choose the canary group sizes. For example, if we instead construct Group A with 10 canaries and Group B

Defense Setting	Group-Privacy ε_{lb}
No Defense	4.43
Defense	27.26

Table 4: Group-privacy ε_{lb} under the gradient canceling attack on MNIST.

with 5 canaries, even if we only allow filtering 1 canary across the whole dataset per epoch, we can still discard both canary groups from the training data, yielding $\varepsilon_{\text{lb}} = 0$.

We also attempt to reproduce this gradient canceling attack within our intended hidden state, input space threat model by leveraging the structure of a single-layer MLP, where sample gradients are aligned with sample features by the chain rule. This suggests that, in principle, one could construct input space canaries whose gradients mimic the desired cancellation behavior; the key challenge is to simulate the desired magnitude, not just the direction, of a gradient space attack in input space.

To this end, we explore several input space constructions for Group A: (i) one-hot vectors with L_∞ norm exceeding the maximum per-sample gradient norm observed across all epochs on a holdout set, (ii) dense random vectors with L_2 norm similarly exceeding this maximum, and (iii) adversarially optimized inputs obtained via FGSM to explicitly induce large gradients. However, these input space constructions fail to replicate the gradient space attack. The core issue is that we cannot reliably control the per-sample gradient magnitudes across training epochs. As a result, only a small fraction of canaries (1%) are consistently filtered, which is insufficient to disrupt the cancellation effect.

D Disparate Impact

A common concern for any filtering-based algorithm is disparate impact. We evaluate our filtering defense on the ColoredMNIST dataset [ABGL19], a variant of MNIST with 2 classes (class 0 has even digits and class 1 has odd digits) where $p\%$ of evens and $(1-p)\%$ of odds are colored red; $(1-p)\%$ of evens and $p\%$ of odds are colored blue. This creates symmetric majority and minority subgroups within both classes. Due to its distribution of relevant (shape) and irrelevant (color) features, balanced by group ratios, ColoredMNIST is a popular synthetic benchmark in the algorithmic fairness and spurious correlations literature [SKHL20, ZSZ$^+$22].

We consider ColoredMNIST under the following values of p: $99.5\%, 95\%, 90\%, 85\%, 80\%, 75\%$. For each instantiation of ColoredMNIST, we evaluate how many samples are filtered by our defense from each of the 4 subgroups, and the resulting impact on groupwise model utility.

Table 5 indicates that our defense does target minority groups over majority groups when marking samples for removal, yielding a disparate reduction in minority group utility. This targeting effect becomes more benign as the fraction p becomes more mild, as expected.

Moreover, we find that the observed bias towards minority group removal is in some sense definitional from the perspective of sample memorization, which any method with a low auditable ε_{lb} must combat. This finding is thematically consistent with other works investigating tradeoffs between privacy and fairness [BPS19, UNK$^+$21]. To formalize this correlation, we audit ColoredMNIST (at $p = 75\%$) without our defense twice, once with the canary as a member of a minority subgroup and once again with the canary as a member of a majority subgroup. Under the minority-group canary, we observe $\varepsilon_{\text{lb}} = 1.145$ whereas under the majority-group canary, we observe $\varepsilon_{\text{lb}} = 0.043$, indicating that minority-group canaries are indeed more prone to memorization.

Table 5: Defense Removal Rate (%): mean $\pm$ std over repetitions

majority_pct	c0_red (maj)	c0_blue (min)	c1_red (min)	c1_blue (maj)
0.995	$1.2 \pm 0.0\%$	$100.0 \pm 0.0\%$	$100.0 \pm 0.0\%$	$1.1 \pm 0.0\%$
0.950	$0.3 \pm 0.0\%$	$27.6 \pm 0.2\%$	$21.4 \pm 0.2\%$	$0.6 \pm 0.0\%$
0.900	$0.5 \pm 0.0\%$	$12.2 \pm 0.1\%$	$8.6 \pm 0.1\%$	$0.8 \pm 0.0\%$
0.850	$0.6 \pm 0.0\%$	$7.9 \pm 0.1\%$	$5.4 \pm 0.1\%$	$0.9 \pm 0.0\%$
0.800	$0.7 \pm 0.0\%$	$5.7 \pm 0.1\%$	$4.1 \pm 0.1\%$	$1.0 \pm 0.0\%$
0.750	$0.8 \pm 0.0\%$	$4.3 \pm 0.1\%$	$3.2 \pm 0.1\%$	$1.1 \pm 0.0\%$

E Alternative Input Space Audits

E.1 Single run audit

A recent work [SNJ23] provides a computationally cheaper alternative to LiRA-style audits by requiring only a single model training run instead of $2T$ shadow models (T being the number of repetitions). This audit measures individual privacy by generating N canaries and independently selecting M of them at random for inclusion in the training dataset. After training, we compute a score (cross entropy loss) for each of the original N canaries. We then make k_+ guesses about which canaries were included in the training and k_- guesses about which were excluded, abstaining from guessing on the remaining canaries. The TPR and FPR from these guesses are used to compute an ε_{lb} via the f-DP framework. We evaluate our defense under the multi-canary individual privacy audit using 500 randomly mislabeled canaries in CIFAR-10. We perform a grid search over all pairs (k_+, k_-) and report the maximum ε_{lb} achieved (Table 6). With our defense, we filter out approximately 75% of the canaries, substantially reducing the ε_{lb}.

E.2 Group privacy audit

We extended LiRA to group privacy by using the maximum loss across all canaries as the audit score for distinguishing between models trained versus without the canary group. We evaluate our defense under this threat model using $k = 500$ mislabeled canaries on CIFAR-10 (Table 6). We evaluate privacy using k-group privacy ε_{lb}; as discussed in Appendix B.1, this value does not formally imply a 1-group privacy ε_{lb} that is $k\times$ smaller, but this is a reasonable scaling to expect. We find that even in the group privacy setting, our defense substantially reduces ε_{lb}.

Experiment	**Train Acc (%)**	**Test Acc (%)**	ε_{lb}
Mislabeled Single Run Audit, No Defense	75.62	70.81	0.20
Mislabeled Single Run Audit, Defense	74.73	70.42	0.10
Mislabeled Group Privacy Audit, No Defense	76.13	71.65	3.54
Mislabeled Group Privacy Audit, Defense	74.69	70.62	2.67

Table 6: Empirical privacy loss under individual and group privacy audits.

F Sample Signatures

This experiment tested the sensitivity of the local filtering defense to the choice of scoring function used to identify high-risk samples. Using these scoring functions, we can detect backdoor canaries that spectral methods [TLM18] may fail to identify. For each attack, the scoring function should identify and filter out canaries as early as possible in training. Filtering canaries too late is ineffective since they have already been memorized. Future work could address late-stage canary detection through privatized gradient ascent.

On CIFAR-10, we considered the following scoring functions.

1. Gradient cosine similarity: how aligned sample gradient direction is with the overall parameter update direction from initialization.
2. L_2 gradient norm: magnitude of their clipped per-sample gradient under the L_2 norm. This is computed using both clipped and unclipped gradients.
3. L_∞ gradient norm: magnitude of the clipped per-sample gradient under the L_∞ norm. This is computed using both clipped and unclipped gradients.
4. Gradient kurtosis unclipped: kurtosis of the gradient distribution, emphasizing heavy-tailed or outlier-like gradients.
5. Directional uniqueness: combines gradient magnitude with a proxy for directional "uniqueness" based on how atypical a sample's gradient direction is relative to a running history.
6. Prediction entropy: entropy of the predicted probability distribution, emphasizing high-uncertainty predictions.

Scoring Function	Train Acc (%)	Test Acc (%)	ε_{lb}
Cosine similarity (θ_0)	75.85	71.05	1.11
Grad norm clipped (L_2)	75.14	70.47	1.37
Grad norm clipped (L_∞)	75.28	70.54	0
Grad norm unclipped (L_2)	75.25	70.73	1.19
Grad norm unclipped (L_∞)	75.20	70.58	0
Gradient kurtosis unclipped	75.49	70.62	0
Directional uniqueness	75.26	70.75	1.20
Pred. entropy	75.11	70.71	0.25
Pred. margin	74.97	70.49	1.11
Rand. proj. variability	75.44	70.83	1.18

Table 7: Scoring function ablation on CIFAR-10 (CNN).

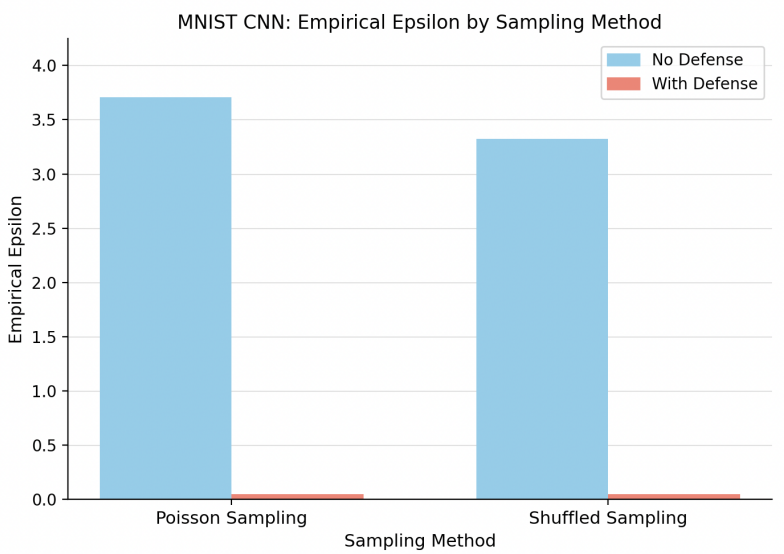


Figure 3: ε_{lb} with/without defense on MNIST (CNN) across Poisson and shuffled sampling methods.

7. Prediction margin: inverted gap between the predicted probability of the true class and the strongest competing class.

8. Random projection variability: projects gradients onto random directions and measures variability in those projections, capturing atypical gradient structure.

All other defense parameters were kept fixed (filtering bandwidth, gradient ascent step, etc.). Across the scoring function choices, the defense consistently detected and removed the canary, with low empirical privacy loss and utility metrics similarly high across configurations (Table 7). This allows practitioners to select or tune the scoring function that best matches their constraints and threat model without changing the defense behavior.

G Poisson vs. Shuffled Subsampling

As discussed in Appendix A.2, in practice, many large-scale implementations of DP-SGD favor shuffled mini-batching over Poisson sampling for efficiency reasons; we therefore verify that our defense remains effective under this more practical sampling scheme. Concretely, we compared the results of estimating ε_{lb} after running our defense versus no defense on a blank canary, for the MNIST dataset, with both Poisson sampling and shuffled subsampling (Figure 3). The results show that the gap in the estimated ε_{lb} persists regardless of sampling strategy.

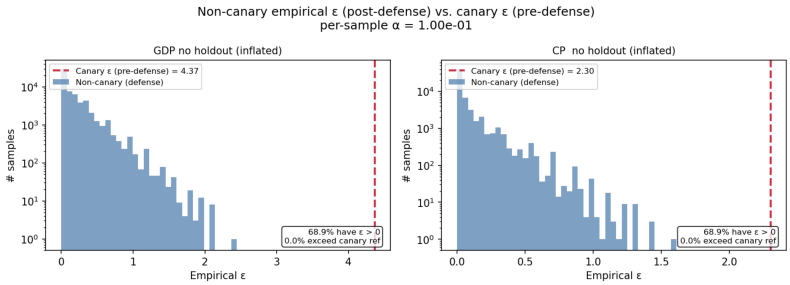


Figure 4: Distribution of empirical ε_{lb} for non-canary training samples (post-defense) vs. canary ε_{lb} (pre-defense) across two auditing methods. In both cases, 0.0% of non-canary samples exceed the canary reference.

H Interpolating Canaries

We linearly interpolated a blank canary with a mislabeled canary with varying degrees of interpolation to test how the effectiveness of the defense scales with canary strength. We constructed canaries as ($\alpha \cdot$ mislabeled sample). When α is 0 the canary is a pure blank image; when α is 1, it is a fully mislabeled image. We evaluated the setup on both MNIST and CIFAR-10 (Table 8).

We found that varying α did not produce a clear change in privacy loss. This suggests that privacy loss is not necessarily proportional to the "strength" of the signal, and the defense remained effective across the different canary constructions.

Dataset	Defense Setting	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1.0$
CIFAR-10	No Defense	0.46	0.42	0.30	0.35	0.73
CIFAR-10	Defense	0	0	0	0.11	0
MNIST	No Defense	2.45	2.45	2.50	2.45	2.50
MNIST	Defense	0.59	0.58	0.62	0.71	0.69

Table 8: Empirical ε_{lb} across interpolated canary strengths on CIFAR-10 and MNIST (CNN).

I The Privacy Onion Effect: Do We Expose Anyone Else?

Prior work [CJZ+22] finds that removing the most privacy vulnerable samples from a dataset may increase the vulnerability of retained samples rather than yielding an aggregate privacy improvement. This "privacy onion" effect suggests that empirical privacy can offer a false sense of security and practitioners ought to employ algorithms with formal privacy guarantees rather than optimizing on empirical privacy. This phenomenon is algorithm-specific, so we analyze its presence for our specific filtering-based defense in this section, by auditing *non-canary samples*.

To empirically test our defense, we trained 1000 shadow models (trained with DP-SGD $\varepsilon = 10$; 500 with the blank canary, 500 without) and recorded per-sample loss under each final model for every sample in the MNIST training dataset. For each non-canary sample, we compute the ε_{lb} post-defense and compare this to the maximum ε_{lb} pre-defense (corresponding to the canary's ε_{lb}). Our results are striking: the (adaptively-chosen) strongest non-canary audit produces a much weaker ε_{lb} than a fixed canary audit. This is true whether or not the formal holdout (false discovery correction) method in Appendix B is used, on both the canary audit and the non-canary audit; in the holdout case, the auditable ε_{lb} of the best non-canary audit drops to 0.

Our audited ε_{lb} comparing the canary with the strongest non-canary sample are displayed in Figures 4 and 5. The experiments in these figures respectively apply the multiple hypothesis heuristic from Appendix B, and use a holdout set to estimate ε_{lb}, for all audits.

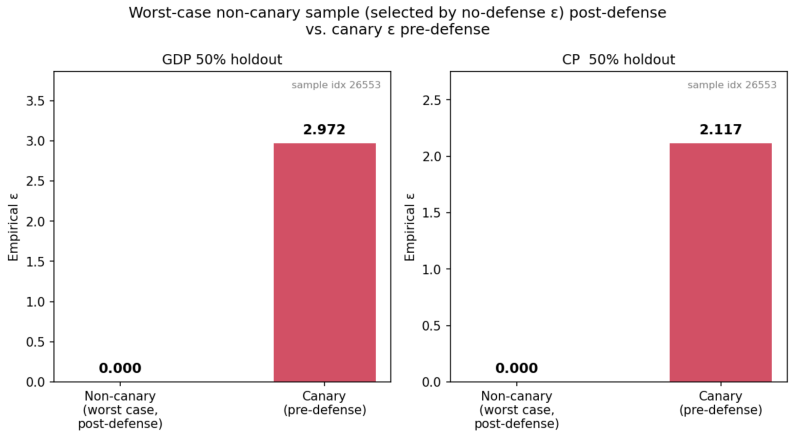


Figure 5: Empirical ε_{lb} of the worst-case non-canary sample post-defense vs. canary ε_{lb} pre-defense, under GDP 50% holdout and CP 50% holdout. The worst-case non-canary sample is selected as the sample achieving the highest empirical ε_{lb} under no defense; its empirical ε_{lb} drops to 0.000 post-defense under both auditing methods.

J Increasing the Trial Count

Our primary experiments used 400 shadow models (200 with the canary included, 200 without) to balance computational constraints. However, the Clopper-Pearson confidence intervals used to compute empirical privacy bounds are looser with fewer trials. We conducted a scaled-up experiment on MNIST with a blank canary using 1000 shadow models (500 per condition) instead of the standard 400. We compared the ε_{lb} and model utility between defense and non-defense settings under the higher sample regime (Table 9). The ε_{lb} gap with and without the defense remained consistent with the 400 trial experiments.

Defense Setting	Train Acc (%)	Test Acc (%)	ε_{lb} (400 Trials)	ε_{lb} (1000 Trials)
No Defense	98.80	98.64	3.71	4.01
Defense	97.60	97.89	0	0.09

Table 9: Empirical privacy loss on MNIST (CNN) with 1000 shadow models.

K Additional Hyperparameter Tuning

K.1 Global Filtering Ablation

We conducted experiments on MNIST using a blank input space canary to evaluate how global filtering (selecting the top-k samples across the entire dataset) compares with local filtering (selecting the top-k samples within each class). Our experiments bias towards local filtering to align with prior work [TLM18], which found that backdoored data poisons exhibit a stronger signature when compared relative to other samples in the same class rather than the entire dataset. We tested three global filtering bandwidths: $k \in \{10, 25, 50\}$ samples filtered per epoch (Table 10). For comparison, our default per-class filtering with $k = 5$ filters, for 50 samples per epoch in total (5 samples per class across 10 classes).

Global filtering achieved similar empirical privacy and test accuracy compared to per-class filtering across all bandwidth configurations. These results suggest that practitioners can choose between filtering strategies based on implementation convenience or specific dataset characteristics rather than privacy-utility tradeoffs alone. For instance, a model trainer may prefer global filtering to limit the total number of discarded samples, or local filtering when class imbalance is a concern. Future work could explore adaptive scheduling approaches that that monitor training dynamics and switch between global and local filtering on a per-epoch basis.

Global Filtering k Value	Train Acc (%)	Test Acc (%)	ε_{lb}
10	0.984	0.984	0
25	0.980	0.981	0.015
50	0.975	0.977	0.090

Table 10: Global filtering ablation on MNIST (CNN).

Local Filtering k Value	Train Acc (%)	Test Acc (%)	ε_{lb}
1	76.01	71.07	0.44
2	75.92	71.11	0.52
3	75.52	70.62	0.43
4	75.28	70.60	0.12
5	75.27	70.77	0.36

Table 11: Local filtering bandwidth ablation on CIFAR-10 (CNN).

K.2 Varying Local Filtering Defense Bandwidth Ablation

To empirically evaluate how the bandwidth parameter k (the number of samples filtered per class per epoch) affects the defense efficacy, we conducted experiments on CIFAR-10 using a blank canary with gradient-norm based local filtering. We tested bandwidth values $k \in \{1, 2, 3, 4, 5\}$ (Table 11). The defense reduced empirical privacy loss across all bandwidth configurations, though the degree of reduction varied with k. These results suggest that the choice of k does have some effect on empirical privacy, and practitioners should consider tuning it alongside computational and utility constraints.

K.3 Varying Filter Frequency Ablation

In this experiment, we varied the filtering frequency during training, applying the defense every k epochs for $k \in \{1, 5, 10, 20\}$. Experiments used the CIFAR-10/CNN architecture with a blank input space canary. We observed that filtering frequency has a meaningful impact on empirical privacy outcomes, with more frequent filtering achieving the lowest εlb of 0.43, while less frequent filtering led to substantially higher loss, peaking at 1.39 when filtering every 10 epochs (Table 12). These results suggest that practitioners should prefer more frequent filtering when empirical privacy is a priority, though utility metrics remained stable across all configurations. Filtering frequency may become a more critical hyperparameter for complex canaries or auditing threat models, where model trainers seek to limit the number of discarded samples during training. Future work could explore adaptive scheduling approaches that monitor training dynamics and apply heuristics to determine when filtering is necessary on a per-epoch basis.

L Full Empirical Privacy Results Across Audit Methods

In all experiments throughout the paper, we report ε_{lb} using the GDP no holdout auditing method as our primary metric (cf. Appendix B). For completeness, here we report full results across all auditing configurations. For GDP, we vary the fraction of shadow models held out for threshold selection at 25%, 50%, and 75%. For the CP (Clopper-Pearson)-based ε_{lb} method, we include results with no holdout as well as the same three holdout levels. These additional metrics provide a more comprehensive picture of the empirical privacy landscape across all experiments, and allow the reader to assess the robustness of our conclusions to the choice of auditing method.

Dataset (Model)	ε_{ub}	Setting	25%	50%	75%	no holdout
MNIST (CNN)	2	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—

Continued on next page

Table 13 – continued from previous page

Dataset (Model)	ε_{ub}	Setting	25%	50%	75%	no holdout
		Defense (CP)	0.000	0.000	0.000	0.015
	4	No Defense (GDP)	0.000	0.000	0.011	—
		No Defense (CP)	0.000	0.000	0.000	0.398
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	6	No Defense (GDP)	0.575	0.000	0.688	—
		No Defense (CP)	0.000	0.000	0.138	0.600
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	8	No Defense (GDP)	1.054	0.424	1.851	—
		No Defense (CP)	0.000	0.000	0.544	0.840
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	10	No Defense (GDP)	0.773	1.580	2.708	—
		No Defense (CP)	0.071	0.184	0.669	1.066
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
Purchase (MLP)	2	No Defense (GDP)	0.569	0.183	0.000	—
		No Defense (CP)	0.028	0.000	0.000	0.164
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	4	No Defense (GDP)	0.000	0.984	0.493	—
		No Defense (CP)	0.002	0.155	0.000	0.739
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.045
	6	No Defense (GDP)	0.008	1.330	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.465
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.025
	8	No Defense (GDP)	0.000	0.000	0.496	—
		No Defense (CP)	0.000	0.000	0.000	0.961
		Defense (GDP)	0.000	0.197	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.132
	10	No Defense (GDP)	0.000	2.047	1.205	—
		No Defense (CP)	0.000	0.397	0.314	0.753
		Defense (GDP)	1.281	1.278	0.902	—
		Defense (CP)	0.205	0.263	0.000	0.503
CIFAR-10 (CNN)	2	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.017
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	4	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	6	No Defense (GDP)	0.000	0.000	0.233	—
		No Defense (CP)	0.000	0.000	0.000	0.477
		Defense (GDP)	0.000	0.000	0.055	—
		Defense (CP)	0.000	0.000	0.000	0.232

Continued on next page

Table 13 – continued from previous page

Dataset (Model)	ε_{ub}	Setting	25%	50%	75%	no holdout
	8	No Defense (GDP)	0.000	0.523	0.931	—
		No Defense (CP)	0.000	0.000	0.194	0.570
		Defense (GDP)	0.000	0.000	0.074	—
		Defense (CP)	0.000	0.000	0.000	0.243
	10	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.570
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
CIFAR-10 (WideResNet-16)	2	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	4	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.145
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	6	No Defense (GDP)	0.000	0.014	0.479	—
		No Defense (CP)	0.000	0.000	0.071	0.407
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	8	No Defense (GDP)	0.150	0.277	0.625	—
		No Defense (CP)	0.000	0.000	0.084	0.619
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	10	No Defense (GDP)	0.516	0.613	0.767	—
		No Defense (CP)	0.000	0.078	0.119	0.671
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.152
CIFAR-10 (CNN+AM4)	2	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.006
	4	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	6	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.026
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	8	No Defense (GDP)	0.000	0.034	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.600
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	10	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.000
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000

Continued on next page

Table 13 – continued from previous page

Dataset (Model)	$\varepsilon_{\mathrm{ub}}$	Setting	25%	50%	75%	no holdout

Table 13: Full empirical privacy results across privacy budgets $\varepsilon_{\mathrm{ub}} \in \{2, 4, 6, 8, 10\}$, datasets, and model architectures. See Section 3.1.

Epochs Before Filtering	Train Acc (%)	Test Acc (%)	ε_{lb}
1	75.21	70.77	0.43
5	75.74	70.73	1.21
10	76.19	71.13	1.39
20	76.06	70.89	1.19

Table 12: Effect of filtering frequency on CIFAR-10 (CNN).

Scoring Function	Setting	25%	50%	75%	no holdout
Cosine similarity (θ_0)	Defense (GDP)	0.000	0.000	0.063	—
	Defense (CP)	0.000	0.000	0.000	0.357
Grad norm clipped (L_2)	Defense (GDP)	0.000	0.394	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.477
Grad norm clipped (L_∞)	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
Grad norm unclipped (L_2)	Defense (GDP)	0.000	0.000	0.023	—
	Defense (CP)	0.000	0.000	0.000	0.391
Grad norm unclipped (L_∞)	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
Gradient kurtosis unclipped	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
Directional uniqueness	Defense (GDP)	0.000	0.394	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.415
Pred. entropy	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
Pred. margin	Defense (GDP)	0.000	0.137	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.378
Rand. proj. variability	Defense (GDP)	0.000	0.101	0.210	—
	Defense (CP)	0.000	0.000	0.000	0.356

Table 14: Scoring function ablation on CIFAR-10 (CNN): full empirical privacy results across auditing methods. See Appendix F.

Dataset	Setting	25%	50%	75%	no holdout
MNIST (CNN)	No Defense (GDP)	1.856	2.313	2.867	—
	No Defense (CP)	0.133	0.891	1.157	1.437
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 15: Empirical privacy loss on MNIST (CNN) with 1000 shadow models, full results across auditing methods. See Appendix J.

Sampling Method	Setting	25%	50%	75%	no holdout
Shuffled	No Defense (GDP)	0.000	0.617	1.335	—
	No Defense (CP)	0.000	0.025	0.368	1.372
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
Poisson	No Defense (GDP)	0.773	1.580	2.708	—
	No Defense (CP)	0.071	0.184	0.669	1.066
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 16: ε_{lb} with/without defense on MNIST (CNN) across Poisson and shuffled sampling methods, full results across auditing methods. See Appendix G.

Audit Type	Dataset	Setting	25%	50%	75%	no holdout
Group privacy (multi-canary)	CIFAR-10	No Defense (GDP)	0.453	1.180	2.563	—
		No Defense (CP)	0.000	0.449	0.701	1.376
		Defense (GDP)	0.000	1.425	1.415	—
		Defense (CP)	0.062	0.394	0.409	1.146

Table 17: Full empirical privacy results under group privacy (multi-canary) mislabeled audits. See Appendix E.2.

Canary Type	Setting	25%	50%	75%	no holdout
Majority canary	No Defense (GDP)	0.000	0.000	0.000	—
	No Defense (CP)	0.000	0.000	0.000	0.000
Minority canary	No Defense (GDP)	0.000	0.000	0.000	—
	No Defense (CP)	0.000	0.000	0.000	0.296

Table 18: Full empirical privacy results for the fairness audit on Colored MNIST, under the no-defense setting. See Appendix D.

Dataset	Setting	25%	50%	75%	no holdout
MNIST (CNN)	No Defense (GDP)	0.153	1.732	2.663	—
	No Defense (CP)	0.199	0.000	0.152	1.052
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
CIFAR-10 (CNN)	No Defense (GDP)	2.845	1.519	1.809	—
	No Defense (CP)	0.000	0.253	0.491	0.767
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 19: Full empirical privacy results under ClipBKD attacks on MNIST and CIFAR-10 (CNN). See Section 3.1.

Dataset	Setting	25%	50%	75%	no holdout
MNIST (CNN)	No Defense (GDP)	0.000	1.452	2.404	—
	No Defense (CP)	0.000	0.340	0.543	1.306
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000
CIFAR-10 (CNN)	No Defense (GDP)	0.018	0.269	0.000	—
	No Defense (CP)	0.000	0.000	0.000	0.362
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 20: Full empirical privacy results under FGSM attacks on MNIST and CIFAR-10 (CNN). See Section 3.1.

Dataset	Setting	25%	50%	75%	no holdout
MNIST (CNN)	No Defense (GDP)	0.701	1.576	1.318	—
	No Defense (CP)	0.065	0.152	0.223	0.806
	Defense (GDP)	0.829	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.124
CIFAR-10 (CNN)	No Defense (GDP)	0.008	0.000	0.000	—
	No Defense (CP)	0.000	0.000	0.000	0.068
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 21: Full empirical privacy results under mislabeled input space attacks on MNIST and CIFAR-10 (CNN). See Section 3.1.

Setting	25%	50%	75%	no holdout
No Defense (GDP)	1.081	2.476	2.410	—
No Defense (CP)	0.150	0.872	0.861	1.760
Defense (GDP)	19.564	24.378	27.130	—
Defense (CP)	2.785	3.493	3.903	4.185

Table 22: Full empirical privacy results under the gradient canceling attack on MNIST (CNN). See Appendix C.

Dataset	α	Setting	25%	50%	75%	no holdout
CIFAR-10	0.00	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.062
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	0.25	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.060
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	0.50	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.024
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	0.75	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.035
		Defense (GDP)	0.000	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
	1.00	No Defense (GDP)	0.000	0.000	0.000	—
		No Defense (CP)	0.000	0.000	0.000	0.159
		Defense (GDP)	0.000	0.113	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.000
MNIST	0.00	No Defense (GDP)	0.701	1.576	1.318	—
		No Defense (CP)	0.065	0.152	0.223	0.806
		Defense (GDP)	0.354	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.102
	0.25	No Defense (GDP)	0.701	1.576	1.247	—
		No Defense (CP)	0.065	0.403	0.210	0.806
		Defense (GDP)	0.109	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.102
	0.50	No Defense (GDP)	0.701	1.590	1.318	—
		No Defense (CP)	0.065	0.403	0.223	0.806
		Defense (GDP)	0.000	0.185	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.145
	0.75	No Defense (GDP)	0.701	1.576	1.318	—
		No Defense (CP)	0.065	0.403	0.223	0.806
		Defense (GDP)	0.354	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.183
	1.00	No Defense (GDP)	0.701	1.590	1.247	—
		No Defense (CP)	0.065	0.403	0.210	0.806
		Defense (GDP)	0.109	0.000	0.000	—
		Defense (CP)	0.000	0.000	0.000	0.169

Table 23: Full empirical privacy results across interpolated canary strengths (α) on CIFAR-10 and MNIST (CNN). See Appendix H.

Epochs Between Filtering	25%	50%	75%	no holdout
1 (GDP)	0.000	0.000	0.000	—
1 (CP)	0.000	0.000	0.000	0.066
5 (GDP)	0.000	0.086	0.302	—
5 (CP)	0.000	0.000	0.013	0.391
10 (GDP)	0.000	0.266	0.000	—
10 (CP)	0.000	0.000	0.000	0.449
20 (GDP)	0.000	0.000	0.000	—
20 (CP)	0.000	0.000	0.000	0.391

Table 24: Full empirical privacy results across filter frequencies on CIFAR-10 (CNN). See Appendix K.3.

Local Filtering k Value	25%	50%	75%	no holdout
1 (GDP)	0.000	0.000	0.000	—
1 (CP)	0.000	0.000	0.000	0.055
2 (GDP)	0.000	0.000	0.000	—
2 (CP)	0.000	0.000	0.000	0.102
3 (GDP)	0.000	0.000	0.000	—
3 (CP)	0.000	0.000	0.000	0.065
4 (GDP)	0.000	0.000	0.000	—
4 (CP)	0.000	0.000	0.000	0.000
5 (GDP)	0.000	0.000	0.000	—
5 (CP)	0.000	0.000	0.000	0.035

Table 25: Full empirical privacy results across local filtering bandwidth values on CIFAR-10 (CNN). See Appendix K.2.

Attack	Setting	25%	50%	75%	no holdout
Mislabeled	No Defense (GDP)	0.701	1.576	1.318	—
	No Defense (CP)	0.065	0.152	0.223	0.806
	Defense (GDP)	0.849	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.153
Defense-aware	No Defense (GDP)	0.000	0.672	1.346	—
	No Defense (CP)	0.000	0.106	0.333	0.531
	Defense (GDP)	0.000	0.000	0.000	—
	Defense (CP)	0.000	0.000	0.000	0.000

Table 26: Full empirical privacy results for the defense-aware audit on MNIST (CNN), comparing standard mislabeled canaries against the fixed-point attack. See Section 3.2.

Global Filtering k Value	25%	50%	75%	no holdout
10 (GDP)	0.000	0.000	0.000	—
10 (CP)	0.000	0.000	0.000	0.000
25 (GDP)	0.000	0.000	0.000	—
25 (CP)	0.000	0.000	0.000	0.000
50 (GDP)	0.000	0.000	0.000	—
50 (CP)	0.000	0.000	0.000	0.000

Table 27: Full empirical privacy results for global filtering ablation on MNIST (CNN). See Appendix K.1.

Setting	25%	50%	75%	no holdout
No Defense (GDP)	17.625	24.378	27.130	—
No Defense (CP)	2.749	3.493	3.903	4.185
Defense (GDP)	17.625	24.378	25.311	—
Defense (CP)	2.749	3.493	3.892	4.185

Table 28: Full empirical privacy results under the gradient bandwidth attack on MNIST (CNN). See Appendix C.

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 claims are clearly stated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations are mentioned throughout, particularly in discussion in the introduction and explorations in the many appendices.

   Guidelines:

   - The answer [N/A] 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: All theoretical results describe any requisite assumptions and are accompanied by complete proofs.

Guidelines:

- The answer [N/A] 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: Code for our experiments is provided.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: Code for our experiments is provided.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are provided in a combination of appendices as well as the released code.

   Guidelines:

   - The answer [N/A] 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: All experiments have uncertainty quantified via Clopper-Pearson confidence intervals which average over hundreds of runs.

   Guidelines:

   - The answer [N/A] 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: Compute resources used are described in the experiments section.

   Guidelines:

   - The answer [N/A] 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: There are no human subjects in the research. All datasets used are publicly available and have no known ethical concerns. Exploration of impacts on fairness were performed and discussed in Appendix D.

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: Section 4 and Appendix D discuss both positive (e.g., a better understanding of privacy research) and negative (e.g., a possible false sense of security, impacts on fairness, etc.) societal impacts, along with corresponding experiments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] 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: All datasets are cited, and licenses are mentioned when known.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: There was no research involving crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: There was no research involving crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used in any important, original, or non-standard way.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.