Privacy Backdoors and the Tightness of DP-SGD

Divyam Anshumaan, Ronak Chauhan, Sarthak Choudhary

May 2, 2026

Abstract

DP-SGD is the standard mechanism for differentially private fine-tuning, with privacy guarantees from the subsampled Rényi DP accountant. In practice, empirical audits of models fine-tuned from public pretrained weights certify only a small fraction of the leakage these guarantees permit, and the resulting slack is what justifies the loose privacy budgets used in production deployments. We provide a self-contained exposition of Feng and Tramèr (2024), who show that the slack closes once the pretrained initialization is itself adversarial. They introduce *privacy backdoors*: malicious pretrained weights that imprint fine-tuning examples into a small attacker-known coordinate set, so that an end-to-end attacker can recover them from the released model. We derive a hypothesis-testing lower bound on the realized privacy parameter that applies to any DP mechanism. Under the backdoor, this bound reduces the analysis of DP-SGD to a one-dimensional Gaussian distinguishing problem and matches the analytical upper bound exactly under an idealized version of the construction. Two CIFAR-10 experiments validate both directions: a backdoored 3-layer MLP fine-tuned with vanilla SGD recovers 56 of 64 captured training images, and an empirical audit of DP-SGD across four target privacy budgets certifies $64 - 72\%$ of the analytical budget. The slack observed in benign end-to-end DP-SGD audits is an artifact of trusted initialization, not a structural property of the privacy analysis.

1 Introduction

Sharing and fine-tuning pretrained models has become a default practice in modern machine learning. Public repositories such as Hugging Face host hundreds of thousands of models, and practitioners routinely download these weights, attach a fresh classification head, and fine-tune on their own data. The datasets used for fine-tuning frequently contain sensitive information, such as medical records, user conversations, or internal communications. It is therefore standard practice to analyze the privacy properties of the resulting model and, increasingly, to adopt differentially private training procedures such as DP-SGD (Abadi et al., 2016).

Two independent classes of risk have been studied in the fine-tuning pipeline. The first is a **supply-chain** risk that stems from the source of the pretrained weights: a malicious provider can tamper with a model's weights so that it misbehaves on specific inputs, compromising the model's *integrity* (Gu et al., 2017; Liu et al., 2018; Hong et al., 2022). The second is a **memorization** risk that stems from the training procedure itself: neural networks trained (or fine-tuned) on sensitive data encode non-trivial information about individual examples, enabling membership inference (Shokri et al., 2017) and data extraction (Carlini et al., 2021). DP-SGD is the canonical defense against the second risk, bounding the per-example influence of the final weights, and is typically deployed under the implicit assumption that the initialization is benign.

Privacy backdoors. Feng and Tramèr (2024) introduce *privacy backdoors*, which bridge the integrity and privacy threats described above. They show that a malicious provider can tamper with a pretrained model's weights so that, during ordinary fine-tuning on the victim's private dataset, individual training examples are imprinted into the weights in a form that the attacker can later recover. The construction relies on a novel primitive, which the authors term a *data trap*: a small number of units are configured so that the gradient update induced by a specific training example is written with high fidelity into an attacker-chosen set of coordinates, and so that no subsequent update overwrites the signal. An adversary with access only to the final fine-tuned model can then read off the captured example directly from those coordinates.

Implications for DP-SGD. The most consequential aspect of the attack is arguably not the reconstruction of the individual examples but its bearing on *differential privacy*. The privacy analysis of DP-SGD rests on a worst-case condition: the per-example gradient differs maximally, on some coordinate, between the neighboring datasets (Abadi et al., 2016). The induced per-step bound is tight for the Gaussian mechanism; its composition across training steps, however, is widely believed to be pessimistic in practice, as the corresponding adversary is assumed to observe intermediate noisy gradient (Nasr et al., 2021). In contrast, a realistic *end-to-end* adversary, with access only to the deployed fine-tuned model, is expected to extract substantially less information than this bound permits; an intuition that has motivated the use of comparatively loose privacy budgets in the production deployments (e.g., $\varepsilon \geq 8$, or $\varepsilon \approx 9$ in Ramaswamy et al. (2020)).

Privacy backdoors invalidate this intuition. The data-trap primitive can be specialized so that, at every DP-SGD step, the gradient of the target example concentrates entirely on a small, attacker-known index set, with ℓ_2-norm equal to the clipping threshold, while the gradients of all other examples vanish on that set. This constitutes precisely the per-step worst case assumed by the analysis. Feng and Tramèr (2024) shows analytically that, under this construction, an end-to-end adversary observing only the final weights can derive a lower bound on the realized ε that closely matches the provable upper bound. The discrepancy between the theoretical and empirical privacy of end-to-end DP-SGD is thus not attributable to structural slack in the analysis; it is a consequence of benign initializations and disappears under an adversarial one.

Scope. This report provides a self-contained exposition of Feng and Tramèr (2024), organized around two questions.

(i) **Construction.** How can an adversary engineer a pretrained initialization whose weights capture individual fine-tuning examples and preserve the captured signal across many SGD steps with gradient clipping and Gaussian noise?

(ii) **Tightness.** Why does such an adversarial initialization render the end-to-end privacy leakage of DP-SGD nearly tight against its provable upper bound, and what is the resulting closed-form lower bound on the realized ε?

2 Problem Statement

This section formalizes the setting in which privacy backdoors operate. We describe the fine-tuning pipeline and fix notation (Section 2.1), specify the adversary's capabilities and the victim's training procedure (Section 2.2), and state the adversary's formal objectives (Section 2.3).

2.1 Setup

We consider the standard fine-tuning pipeline for supervised classification. A model provider publishes a pretrained backbone $\theta_0^{\text{bb}} \in \mathbb{R}^{d_{\text{bb}}}$. A victim downloads these weights, appends a freshly initialized classification head $\theta_0^{\text{head}} \in \mathbb{R}^{d_{\text{head}}}$ drawn from a standard initializer, and forms the initial parameter vector $\theta_0 := (\theta_0^{\text{bb}}, \theta_0^{\text{head}}) \in \mathbb{R}^d$ with $d = d_{\text{bb}} + d_{\text{head}}$. The victim then fine-tunes on a private dataset $D = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the input space and $\mathcal{Y} = \{1, \ldots, C\}$ is the label space of C-way classification task. Fine-tuning is performed by a training algorithm $\mathcal{A}$, producing the fine-tuned weights $\theta_T = \mathcal{A}(\theta_0, D)$. The adversary in our setting is the model provider itself: having controlled the backbone coordinates of θ_0, it subsequently obtains access to θ_T.

2.2 Threat Model

Adversary. The adversary fully controls the pretrained backbone θ_0^{bb}, subject only to a utility constraint: the fine-tuned model must remain sufficiently accurate on the downstream task that the victim accepts and deploys it. The adversary does not participate in the fine-tuning process and observes no intermediate state of the run, i.e., no per-step gradients. Its only interaction with the fine-tuning pipeline is through the final model θ_T produced by the victim.

Victim. The victim performs standard supervised fine-tuning on the private dataset D using a fixed training algorithm $\mathcal{A}$. We consider two instantiations of $\mathcal{A}$, corresponding to the non-private and differentially private regimes:

- *Full fine-tuning with SGD.* All coordinates of θ_0 are updated via mini-batch stochastic gradient descent for T steps. This is the setting of the reconstruction attack of Section 3.
- *Top-k fine-tuning with DP-SGD.* The bottom portion of the backbone is frozen, and only the top k layers together with the classification head are updated, using the DP-SGD algorithm of Abadi et al. (2016). This is the standard setup for differentially private transfer learning (Tramer and Boneh, 2020), and is the setting under which the tightness of consequence of Section 4 is established.

In both regimes, fine-tuning is carried out end-to-end by the victim, without any adversarial participation.

Access to the fine-tuned model. We restrict attention to the *white-box* setting, in which the adversary reads the fine-tuned parameter vector θ_T directly. This models a fine-tuned model that is redistributed internally or externally, or one whose weights are otherwise recovered by the adversary after deployment. We note in passing that the attack admits a black-box extension in which the adversary only queries the deployed model on inputs of its choice; the black-box setting is not pursued in this report.

2.3 Objective

The adversary's objective has two components: a privacy compromise, formalized as the ability to reconstruct individual fine-tuning examples from the released model; and a utility constraint, formalized as a bound on the degradation of the downstream task performance relative to a benign pretrained initialization.

For the reconstruction goal, we assume the input space $\mathcal{X}$ is equipped with a dissimilarity measure $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ (for instance, the Euclidean distance on normalized images, or a token-level edit

distance on text). Let $\mathcal{P}$ denote a distribution over datasets of size n in $\mathcal{X} \times \mathcal{Y}$, from which the victim draws the private dataset D.

Definition 1 (Reconstruction)**.** *A pair* (θ_0, Ext)*, consisting of an initialization* $\theta_0 \in \mathbb{R}^d$ *and a (possibly randomized) extraction algorithm* $\mathrm{Ext} : \mathbb{R}^d \to \mathcal{X}^*$*, achieves* (k, τ, β)-reconstruction *under the training algorithm* $\mathcal{A}$ *and dataset distribution* $\mathcal{P}$ *if*

$$\mathbb{P}\left[\left|\left\{ (x,y) \in D \ : \ \min_{\hat{x} \in \mathrm{Ext}(\theta_T)} \rho(x, \hat{x}) \ \leq \ \tau \right\}\right| \ \geq \ k \right] \ \geq \ 1 - \beta,$$

where $D \sim \mathcal{P}$*,* $\theta_T = \mathcal{A}(\theta_0, D)$*, and the probability is taken over the randomness of* D*,* $\mathcal{A}$*, and* Ext*.*

Definition 2 (Utility preservation)**.** *Let* θ_0^{ref} *be a benign reference initialization (with a clean pretrained backbone). For a parameter vector* $\theta \in \mathbb{R}^d$ *and a dataset* $D' \subset \mathcal{X} \times \mathcal{Y}$*, let*

$$\mathrm{Acc}(\theta; D') \ := \ \frac{1}{|D'|} \sum_{(x,y) \in D'} \mathbb{1}[\, f_\theta(x) = y \,]$$

denote the empirical accuracy of the classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ *on* D'*. initialization* θ_0 *is* γ-utility-preserving relative to θ_0^{ref} *under the training algorithm* $\mathcal{A}$ *and distribution* $\mathcal{P}$ *if*

$$\mathbb{E}\Big[\mathrm{Acc}\Big(\mathcal{A}(\theta_0^{\mathrm{ref}}, D); D\Big) \ - \ \mathrm{Acc}\Big(\mathcal{A}(\theta_0, D); D\Big)\Big] \ \leq \ \gamma,$$

where the expectation is taken over the randomness of D *and* $\mathcal{A}$*.*

A successful privacy backdoor is an initialization θ_0 paired with an extraction algorithm Ext that meets both objectives simultaneously: the pair achieves (k, τ, β)-reconstruction in the sense of Definition 1 for small τ and β, while θ_0 is γ-utility-preserving in the sense of Definition 2 for small γ.

A consequence for DP-SGD. Definitions 1 and 2 make no reference to differential privacy. When $\mathcal{A}$ is DP-SGD, however, a successful privacy backdoor yields, as a byproduct, an analytic lower bound on the realized privacy parameter ε. This lower bound is a consequence of the reconstruction objective, and we defer its statement and proof to Section 4.

3 Backdooring MLPs for White-box Data Stealing

The goal of the adversary is to have the privacy backdoor capture exactly one finetuning data point. In a federated learning setting, the attacker gets to see gradient updates directly, making it feasible to determine finetuning data points. However, in this setting, the adversary only receives the finetuned model *after* the entire training run is complete. Therefore, the backdoor must be engineered in such a way that the captured inputs:

(i) can be extracted from the weights of the finetuned model,

(ii) "survive" the entire training run, without being mixed up with other inputs captured in subsequent training steps.

To accomplish both criteria, the authors propose *data trap*, a backdoor with a "single-use" property; once the backdoor has been activated, it will never be active again afterwards.

3.1 Data Traps

We illustrate the approach by backdooring a single linear unit (i.e., one element of a linear layer) and then work our way to a full MLP. Let $\mathbf{w} \in \mathbb{R}^m$ and $b \in \mathbb{R}$ be the weights and bias of the unit. Suppose $\mathbf{x} \in \mathbb{R}^m$ is the input of the unit. We define the unit's activation h as,

$$h = \texttt{ReLU}\big(\mathbf{w}^\top \mathbf{x} + b\big) \tag{1}$$

As described in Fowl et al. (2022), the parameters $\mathbf{w}$ and b can be set to ensure that, with high probability, a single input $\mathbf{x}$ in a training batch activates the neuron (i.e. $h > 0$). When the weights update by backpropagating the training loss $\mathcal{L}$, we then get:

$$\nabla_{\mathbf{w}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial h} \cdot \mathbf{x}, \qquad \nabla_b \mathcal{L} = \frac{\partial \mathcal{L}}{\partial h}. \tag{2}$$

Recovering data. We update the weights and bias of the unit as:

$$\mathbf{w}' \leftarrow \mathbf{w} - \eta \cdot \nabla_{\mathbf{w}} \mathcal{L}, \qquad b' \leftarrow b - \eta \cdot \nabla_b \mathcal{L}. \tag{3}$$

Knowledge of the learning rate, the original weights and bias allows us to recover $\nabla_{\mathbf{w}} \mathcal{L}$ and $\nabla_b \mathcal{L}$. Dividing $\nabla_{\mathbf{w}} \mathcal{L}$ by $\nabla_b \mathcal{L}$ recovers $\mathbf{x}$.

Preventing subsequent updates. We now need to ensure that the gradient update for $\mathbf{x}$ closes the backdoor, i.e., prevents further updates. Assume that the backdoor activates on input $\hat{\mathbf{x}}$. The updated backdoor's output on a new input $\mathbf{x}$ is:

$$h' = \texttt{ReLU}\big(\mathbf{w}'^\top \mathbf{x} + b'\big) \tag{4}$$

$$= \texttt{ReLU}\big[(\mathbf{w}^\top \mathbf{x} + b) - \eta \cdot \frac{\partial L}{\partial h} \cdot \Big(\hat{\mathbf{x}}^\top \mathbf{x} + 1\Big)\big] \tag{5}$$

For the ReLU activation function, a sufficient condition for the backdoor to close (i.e. $h' = 0$ for all $\mathbf{x}$, so all subsequent $\frac{\partial h'}{\partial x} = 0$) is to ensure $\mathbf{w}'^\top \mathbf{x} + b' \leq 0$. This is achieved by:

(i) $\hat{\mathbf{x}}^\top \mathbf{x} + 1 > 0$,

(ii) $\frac{\partial L}{\partial h}$ is sufficiently large and positive.

The first condition $\hat{\mathbf{x}}^\top \mathbf{x} + 1 > 0$ only depends on the input, and can be algorithmically enforced (e.g., by mapping inputs to $[0, 1]^m$, which ensures all dot products are non-negative). We take a closer look at the second condition in the following subsection.

3.1.1 Ensuring a large, positive gradient.

Suppose the new finetuing linear layer is $\mathbf{w}^{(2)} = (w_1^{(2)}, \ldots, w_C^{(2)})$, where C are the number of classes for the finetuning task. It is added in front of the last layer of the pretrained model, as shown in Figure 1. We connect the backdoor output h to a hidden unit h' with weight $w_1^{(1)}$, i.e.,

$$h' = \texttt{ReLU}\Big(w_1^{(1)} h\Big).$$

The unit h' is then connected via the new linear layer $\mathbf{w}^{(2)}$ to the model's logit layer $\mathbf{z}$. The logits are passed through a softmax before computing the cross-entropy loss (for some target class T). Starting from the backdoor, the computation proceeds as follows:

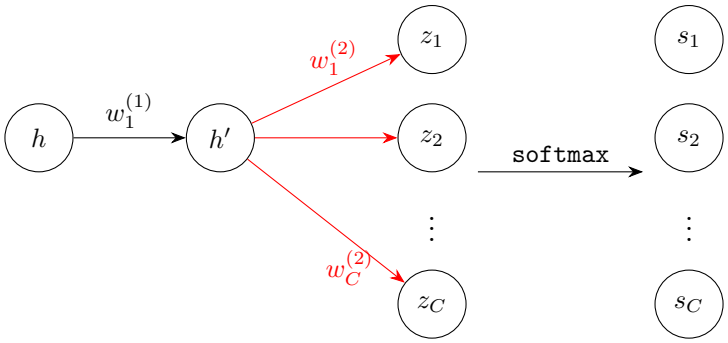

Figure 1: Illustration of the output of a data trap ($h = \texttt{ReLU}\big(\mathbf{w}^\top \mathbf{x} + b\big)$) connected to the model's output. The weights of the classification head (in red) are typically not under the control of the attacker, and randomly initialized before finetuning.

1. Hidden unit activation: $h' = \texttt{ReLU}\Big(w_1^{(1)} h\Big)$ Note that $h > 0$ by construction.
2. Logits: $z_i = w_i^{(2)} h'$ for each class $i \in \{1, \ldots, C\}$
3. Softmax: $s_i = \exp(z_i) / \sum_{j=1}^{C} \exp(z_j)$
4. Cross-entropy loss: $\mathcal{L} = -\sum_{i=1}^{C} y_i \log(s_i)$, where $y_i = \begin{cases} 1 & \text{if } i = T, \\ 0 & \text{otherwise.} \end{cases}$.

A standard gradient calculation gives:

$$\frac{\partial \mathcal{L}}{\partial h} = \frac{\partial \mathcal{L}}{\partial h'} \cdot \frac{\partial h'}{\partial h} = \left(\sum_{i=1}^{C} (s_i - y_i) \cdot w_i^{(2)} \right) \cdot w_1^{(1)} \tag{6}$$

Taking into account the values of y_i, this can be expressed as:

$$\frac{\partial \mathcal{L}}{\partial h} = \left(\sum_{i=1}^{C} \left(s_i \cdot w_i^{(2)} \right) - w_T^{(2)} \right) \cdot w_1^{(1)} \tag{7}$$

In order to ensure $\frac{\partial \mathcal{L}}{\partial h}$ is large, we set $w_1^{(1)}$ to be a large value. Now let's analyze the implications of a large $w_1^{(1)}$. Consider s_i:

$$s_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)} = \frac{\exp(w_i^{(2)} h')}{\sum_{j=1}^{C} \exp(w_j^{(2)} h')}$$

Factorizing $\exp(w_i^{(2)} h')$, we get:

$$s_i = \frac{1}{1 + \sum_{j \neq i}^{C} \exp\left((w_j^{(2)} - w_i^{(2)}) h' \right)} \tag{8}$$

Since $\mathbf{w}^{(2)}$ is randomly initialized, there exists $w_k^{(2)} = \max\{w_1^{(2)}, \ldots, w_C^{(2)}\}$ (ignoring ties). However, $h' = \texttt{ReLU}\Big(w_1^{(1)} h\Big)$ is a large and positive value, since $w_1^{(1)}$ is large and positive, with $h > 0$. Then,

Case 1: $i = k$. Since $w_k^{(2)}$ is the largest value in $\mathbf{w}^{(2)}$, $(w_j^{(2)} - w_i^{(2)}) < 0$ for all $j \neq i$ in Eq. (8). Then for all $j \neq i$, $\exp\big((w_j^{(2)} - w_i^{(2)})\, h'\big) \approx 0$. Therefore,

$$\sum_{j \neq i}^{C} \exp\left((w_j^{(2)} - w_i^{(2)}) h'\right) \approx 0$$

Case 2: $i \neq k$. There will be at least one term $(w_j^{(2)} - w_i^{(2)}) > 0$, where $j = k$. Then for that term, $\exp\big((w_j^{(2)} - w_i^{(2)})\, h'\big) \approx \inf$. Therefore,

$$\sum_{j \neq i}^{C} \exp\left((w_j^{(2)} - w_i^{(2)}) h'\right) \approx \inf$$

Substituting these cases in Eq. (8), we get:

$$s_i \approx \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

Using this knowledge in Eq. (7), we find that:

$$\frac{\partial \mathcal{L}}{\partial h} = \left(\sum_{i=1}^{C} \left(s_i \cdot w_i^{(2)} \right) - w_T^{(2)} \right) \cdot w_1^{(1)} \tag{9}$$

$$\approx \left(w_k^{(2)} - w_T^{(2)} \right) \cdot w_1^{(1)} \tag{10}$$

Now, whenever $T \neq k$, i.e., the target class does not correspond to the largest weight in the linear layer, $\frac{\partial \mathcal{L}}{\partial h} > 0$. The probability of $T = k$ is $1/C$, (as $k \in \{1, \dots C\}$). Therefore, a sufficiently large $w_1^{(1)}$ also guarantees that $\frac{\partial \mathcal{L}}{\partial h}$ is positive with probability $1 - 1/C$. This closes the backdoor and prevents subsequent updates from corrupting the captured inputs.

4 Privacy Analysis

This section establishes the central claim of the report: under a privacy backdoor, the end-to-end privacy analysis of DP-SGD is nearly tight. We first recall DP-SGD and its upper bound, then develop a hypothesis-testing lower bound on ε that applies to any mechanism, and finally instantiate this lower bound under the backdoor construction of Section 3 to show that it matches the upper bound.

4.1 DP-SGD and its privacy upper bound

Fix a per-example loss ℓ, expected lot size L, clipping threshold C, noise multiplier σ, learning rate η, and number of iterations T. DP-SGD (Abadi et al., 2016) iterates, for $t = 1, \dots, T$:

(i) Sample a lot $S_t \subseteq [n]$ by Poisson sub-sampling with rate $q = L/n$;

(ii) Compute per-example gradients and clip to norm C: $g_t(i) = \mathrm{clip}_C(\nabla_\theta \ell(\theta_{t-1}; x_i, y_i))$ for each $i \in S_t$;

(iii) Form the noisy aggregate $G_t = \sum_{i \in S_t} g_t(i) + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma^2 C^2 I_d)$;

(iv) Update $\theta_t = \theta_{t-1} - (\eta/L)\, G_t$.

Theorem 1 (DP-SGD privacy upper bound; Abadi et al., 2016; Mironov et al., 2019)**.** *For any $\delta \in (0,1)$, there exists $\varepsilon^\star(\sigma, q, T, \delta)$, computable from the subsampled-Gaussian Rényi-DP accountant, such that the map $D \mapsto (G_1, \ldots, G_T)$ is $(\varepsilon^\star, \delta)$-differentially private.*

Because θ_T is a deterministic function of θ_0 and $(G_1, \ldots, G_T)$, the post-processing theorem of DP (Dwork and Roth, 2014) implies that end-to-end map $D \mapsto \theta_T$ is also $(\varepsilon^\star, \delta)$-DP. **The question is whether this upper bound is tight when the adversary sees only θ_T.**

4.2 End-to-end versus per-step adversaries

Two adversary models inherit the upper bound $\varepsilon^\star$ via post-processing. A *per-step* adversary observes the full trajectory $(G_1, \ldots, G_T)$; an *end-to-end* adversary observes only θ_T. Empirical auditing shows that $\varepsilon^\star$ is essentially tight in the per-step regime, but audits of end-to-end DP-SGD from benign initializations typically saturate at $\tilde{\varepsilon} \approx \varepsilon^\star/4$ or less (Nasr et al., 2021; Jagielski et al., 2020). This apparent slack is often cited in support of loose production privacy budgets, e.g. $\varepsilon \in [8, 9]$ (Ramaswamy et al., 2020). Whether it reflects structural slack in the analysis, or merely the benign initialization assumed by existing audits, is the question we answer.

4.3 Privacy as hypothesis testing

We first record the reduction from differential privacy to a hypothesis-testing constraint. The reduction is mechanism-agnostic and will be applied to DP-SGD in Section 4.5.

Let $M : \mathbb{D} \to \Delta(\mathcal{R})$ be a randomized mechanism mapping datasets to distributions over an output space $\mathcal{R}$. Two datasets $D_0, D_1 \in \mathbb{D}$ are *neighbors*, written $D_0 \sim D_1$, if they differ by a single record. In the membership-inference specialization to which we will reduce, $D_1 = D \cup \{x^\star\}$ and $D_0 = D$ for a target canary $x^\star$, and the adversary must decide from $Y \sim M(D_b)$ whether $b = 0$ or $b = 1$.

Definition 3 (Deterministic test)**.** *A* test *is specified by a measurable rejection region $R \subseteq \mathcal{R}$: the adversary declares "$x^\star$ present" if $Y \in R$ and "$x^\star$ absent" otherwise. Its power is summarized by*

$$\mathrm{TPR}(R) = \mathbb{P}_{Y \sim M(D_1)}[Y \in R], \qquad \mathrm{FPR}(R) = \mathbb{P}_{Y \sim M(D_0)}[Y \in R].$$

Lemma 2 (Hypothesis-testing constraint)**.** *If M is (ε, δ)-differentially private, then for every rejection region $R \subseteq \mathcal{R}$,*

$$\mathrm{TPR}(R) \;\le\; e^{\varepsilon} \cdot \mathrm{FPR}(R) + \delta.$$

Proof. Apply the definition of (ε, δ)-DP with measurable set $S = R$ to the neighboring pair $D_0 \sim D_1$ and substitute the definitions of TPR and FPR. □

Proposition 3 (Hypothesis-testing lower bound on ε)**.** *For every rejection region R with $\mathrm{FPR}(R) > 0$,*

$$\varepsilon \;\ge\; \log\left(\frac{\mathrm{TPR}(R) - \delta}{\mathrm{FPR}}\right). \tag{11}$$

Consequently, if one defines

$$\tilde{\varepsilon}(M, \delta) \;\coloneqq\; \sup_{R \subseteq \mathcal{R}} \log\left(\frac{\mathrm{TPR}(R) - \delta}{\mathrm{FPR}(R)}\right),$$

then any valid (ε, δ)-DP guarantee for M must satisfy $\varepsilon \ge \tilde{\varepsilon}(M, \delta)$.

Proof. Rearrange Lemma 2 and take the supremum over R. □

4.4 Reduction to a scalar threshold test

Optimizing $\tilde{\varepsilon}$ over $R \subseteq \mathcal{R}$ is intractable for $\mathcal{R} = \mathbb{R}^d$. The standard simplification is to project the output through a scalar test statistic and restrict attention to threshold rejection regions.

Definition 4 (Threshold test)**.** *Given a measurable test statistic $\Delta : \mathcal{R} \to \mathbb{R}$, the* threshold family *is $\{R_t\}_{t\in\mathbb{R}}$ with $R_t = \{y \in \mathcal{R} : \Delta(y) \geq t\}$. Define*

$$\tilde{\varepsilon}_{\text{thr}}(\Delta, \delta) \;:=\; \sup_{t\in\mathbb{R}} \log\left(\frac{\mathbb{P}_{Y\sim M(D_1)}[\Delta(Y) \geq t] - \delta}{\mathbb{P}_{Y\sim M(D_0)}[\Delta(Y) \geq t]}\right).$$

By the Neyman–Pearson lemma, if Δ is a monotone function of the log-likelihood ratio $\Lambda(y) = \log \frac{dM(D_1)}{dM(D_0)}(y)$, then threshold tests on Δ are uniformly most powerful: for every $\alpha \in [0, 1]$ there is a t such that R_t maximizes $\text{TPR}(R)$ subject to $\text{FPR}(R) \leq \alpha$ among *all* tests. In that case $\tilde{\varepsilon}_{\text{thr}}(\Delta, \delta) = \tilde{\varepsilon}(M, \delta)$, i.e., no tightness is lost by restricting to threshold tests. When Λ is not available in closed form, one may replace it with an approximation; the privacy backdoor construction of the next subsection is engineered so that a single coordinate of θ_T is, up to known affine shifts, a sufficient statistic for Λ, and the restriction is therefore without loss. The optimal threshold $t^\star$ is then found by grid search.

4.5 Instantiation under a privacy backdoor

We now instantiate Section 4.4 on DP-SGD applied to a backdoored initialization. Fix a target canary $x^\star$ and consider the neighboring datasets $D_0 = \{(x_i, y_i)\}_{i=1}^{n-1}$ and $D_1 = D_0 \cup \{x^\star\}$. Let $\theta_T = \mathcal{A}_{\text{DP-SGD}}(\theta_0, D_b)$.

The canary construction. The construction of Section 3 can be specialized to produce a canary *module* rather than a single-use reconstruction trap: a configuration of the backbone weights, together with a known coordinate $j^\star$ and known affine shifts $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}_{>0}$, such that the clipped per-example gradient along $j^\star$ satisfies, for every step t,

$$g_t(i)[j^\star] \;=\; \begin{cases} C & \text{if } i = x^\star \text{ and } x^\star \in S_t, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Unlike the latched reconstruction trap of Section 3, the canary module is *multi-use*: (12) holds at every step, not only at the first activation. The latch is unnecessary here because the canary does not encode an input-dependent payload; the attacker knows $x^\star$ a priori, so there is no information that needs to be preserved from being updated. The construction must instead secure two invariants: the clipped per-example gradient on coordinate $j^\star$ equals $\pm C$ when $i = x^\star$ is in the lot and 0 otherwise, and the detector's activation condition on $x^\star$ is preserved across all T steps.

Both invariants are produced by standard architectural choices. The trap unit is tuned so that it activates only for inputs in a small neighborhood of $x^\star$, which forces every $i \neq x^\star$ to contribute zero gradient through the amplifier path, including on coordinate $j^\star$. For $i = x^\star$, an amplifier gain $M \gg 1$ ensures that the pre-clip gradient $\nabla_\theta \ell(\theta_{t-1}; x^\star)$ is dominated by a term of magnitude M along $e_{j^\star}$, so clipping by the factor C/M yields $\pm C$ on coordinate $j^\star$ and $O(C/M)$ on every other coordinate. Because non-target examples leave the detector weights untouched, and the target's own contribution to those weights is $O(C/M)$ per step, the cumulative drift of any detector coordinate across T steps is bounded by $O(\eta T C/(LM)) + O(\eta\sigma C\sqrt{T}/L)$, which is small by choice of M relative to the activation margin with which the detector is configured. The detector, therefore, remains operational throughout training.

Scalar reduction. Define the canary statistic $\Delta(\theta) \coloneqq (\theta[j^\star] - \alpha)/\beta$. Under DP-SGD, combining (12) with the noise addition at coordinate $j^\star$ gives

$$\Delta(\theta_T) \;=\; \sum_{t=1}^{T} \big(C \cdot B_t + Z_t\big), \qquad Z_t \sim \mathcal{N}(0, \sigma^2 C^2) \text{ i.i.d.}, \tag{13}$$

where $B_t = \mathbb{1}[x^\star \in S_t]$. Under D_0, the canary is absent and $B_t \equiv 0$; under D_1, $B_t \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(q)$. Writing $X_t \coloneqq CB_t + Z_t$, equation (13) is the T-fold sum of i.i.d. copies of the subsampled-Gaussian step (Mironov et al., 2019), and the induced marginals are

$$\Delta(\theta_T) \,\big|\, D_0 \;\sim\; \mathcal{N}(0, T\sigma^2 C^2), \tag{14}$$

$$\Delta(\theta_T) \,\big|\, D_1 \;\sim\; C \cdot \text{Bin}(T, q) \;+\; \mathcal{N}(0, T\sigma^2 C^2), \tag{15}$$

where (15) denotes the distribution of the sum of an independent $C \cdot \text{Bin}(T, q)$ draw and a Gaussian with variance $T\sigma^2 C^2$.

Sufficiency and optimality. The likelihood ratio between (15) and (14) depends on θ_T only through $\Delta(\theta_T)$, so Δ is a sufficient statistic for distinguishing D_0 from D_1; moreover, the ratio is monotone non-decreasing in Δ. By Neyman–Pearson, threshold tests on Δ are uniformly most powerful.

Main theorem. The scalar mechanism defined by (13) is, by construction, the subsampled Gaussian mechanism with sampling rate q and noise multiplier σ iterated T times, whose optimal privacy curve is known (Mironov et al., 2019) to match the DP-SGD upper bound of Theorem 1. We therefore obtain:

Theorem 4 (End-to-end tightness under a privacy backdoor)**.** *Let θ_0 be the backdoored initialization of Section 3 with canary module (12). Let $\varepsilon^\star(\sigma, q, T, \delta)$ be the DP-SGD upper bound of Theorem 1, and let $\tilde{\varepsilon}(\sigma, q, T, \delta)$ be the hypothesis-testing lower bound obtained from Proposition 3 via the canary statistic Δ. Then*

$$\tilde{\varepsilon}(\sigma, q, T, \delta) \;=\; \varepsilon^\star(\sigma, q, T, \delta). \tag{16}$$

Proof sketch. By (14)–(15) and the sufficiency argument above, $\tilde{\varepsilon}$ equals the hypothesis-testing lower bound for the T-fold subsampled Gaussian mechanism on $\mathbb{R}$ with rate q and noise σ. The hypothesis-testing lower bound for this mechanism coincides with its f-DP characterization (Mironov et al., 2019; Dong et al., 2022), which is in turn equal to the RDP-accountant upper bound $\varepsilon^\star$. The full calculation proceeds by expressing $\tilde{\varepsilon}$ as a supremum of Gaussian-to-Gaussian-mixture likelihood-ratio integrals and invoking tightness of the subsampled-Gaussian accountant; we refer to Feng and Tramèr (2024) for the detailed numerical treatment. □

4.6 Near-tightness and implications

The equality in Theorem 4 is an idealization; in practice, ε falls slightly below ε^* because the canary module does not activate on every step it is in the lot.

Realized vs. idealized canaries. The identity (12) assumes the canary module is perfectly preserved throughout training. In practice, activation of the canary may fail on a fraction of steps, and the realized per-step activation probability is $q\gamma$ for some $\gamma \in (0, 1]$ close to but not exactly 1. The argument above then yields $\tilde{\varepsilon}(\sigma, q, T, \delta) = \varepsilon^\star(\sigma, q\gamma, T, \delta)$, which remains a large fraction of the upper bound under typical DP-SGD parameters. A representative numerical comparison ($\sigma = 1$, $q = 10^{-2}$, $T = 10^3$, $\delta = 10^{-5}$) gives $\varepsilon^\star \approx 2.3$ and $\tilde{\varepsilon}/\varepsilon^\star \geq 0.87$ for $\gamma \geq 0.9$; a fuller evaluation is deferred to experiments.

Implications. The slack commonly observed in empirical end-to-end audits of DP-SGD from benign initializations is not a structural feature of the privacy analysis. It is a consequence of the initialization being drawn independently of the fine-tuning data. Theorem 4 shows that an adversary who controls the initialization can eliminate this slack and certify a lower bound matching the theoretical upper bound.

5 Experiments

We empirically validate the two main claims of the report on CIFAR-10. Section 5.1 confirms that the data-trap of Section 3 leaks individual training inputs from the released weights of a non-private fine-tuning run. Section 5.2 instantiates the canary reduction of Section 4.5 as a single-canary audit of DP-SGD and shows that the audited $\tilde{\varepsilon}$ recovers a substantial fraction of the analytical privacy budget across $\varepsilon \in \{1, 2, 4, 8\}$.

5.1 Reconstruction under vanilla SGD

We instantiate the data trap of Section 3 on a 3-layer MLP with hidden widths $(256, 256)$, and fine-tune it on 10,000 CIFAR-10 images for 20 epochs of plain SGD with learning rate 0.05 and batch size 64. Of the 256 first-layer units, $k = 64$ are reserved as data traps; the remaining 192 stay at their default initialization. Each trap's input row $\mathbf{w}_j \in \mathbb{R}^{3072}$ is a uniform draw from the unit sphere, and its bias is set to the empirical 0.999-quantile of $\{\mathbf{w}_j^\top x_i\}_i$ on the 10,000-image training subset — so that, in expectation, ten training images cross the trap's threshold. The corresponding amplifier neuron in the second hidden layer carries weight $w_1^{(1)} = 750$ from the trap; the head connection is left at its random initialization. Inputs use only `ToTensor`'s rescaling to $[0, 1]^{3072}$, with no mean/std normalization, ensuring $\hat{\mathbf{x}}^\top \mathbf{x} + 1 > 0$ holds and the latching argument of Section 3 applies.

After fine-tuning, the captured input at trap j is recovered by the single division

$$\hat{x}_j \;=\; \frac{W_T^{(1)}[j,:] \;-\; W_0^{(1)}[j,:]}{b_T^{(1)}[j] \;-\; b_0^{(1)}[j]} \;\in\; \mathbb{R}^{3072},$$

reshaped to $32 \times 32 \times 3$ (Figure 2). 56 of the 64 traps reconstruct a single training image at PSNR ≥ 30 dB and SSIM ≥ 0.97; the median PSNR across all 64 cells is 155.8 dB. The eight failed cells are blurred two- or three-image mixtures (PSNR $\in [21, 27]$ dB), corresponding to traps where multiple of the ten activator images happened to land in the same SGD lot — the "noisy mixture" failure mode anticipated by Feng and Tramèr (2024, §4.3). The amplifier value $w_1^{(1)} = 750$ is large enough to drive the bias far below threshold in a single SGD step, so the second failure mode (latch-fails-and-trap-is-overwritten) does not appear in this run. Our 56/64 recovery rate matches Feng and Tramèr (2024, Figure 2) essentially exactly.

Ground truth Reconstruction $\hat{x}_j = \Delta W_j^{(1)} / \Delta b_j^{(1)}$

Figure 2: Captured CIFAR-10 inputs after 20 epochs of vanilla SGD on a backdoored 3-layer MLP with $w_1^{(1)} = 750$. Cells where exactly one training image crossed the trap's threshold reconstruct at PSNR ≥ 30 dB; gray-tinted cells in the ground-truth grid correspond to traps activated by multiple images, which appear as blurred mixtures in the reconstruction.

5.2 Single-canary audit of DP-SGD

Setup. The architecture is a frozen pretrained CNN encoder $\phi : \mathbb{R}^{3072} \to \mathbb{R}^{1024}$ (8 epochs of plain SGD, val. acc. 55.6%) followed by a 2-layer MLP head with hidden widths $(128, 64)$. A single canary unit at coordinate $j^\star = 127$ of the first MLP layer is configured so that exactly one training image $x^\star$ activates it. We follow Feng and Tramèr (2024)'s protocol: fix $\sigma = 1$, $L = 500$ (sampling rate $q = 10^{-2}$), $C = 1$, $\delta = 10^{-5}$, and vary the number of optimizer steps T per row to reach each target ε.

Choice of T. With σ, q, and δ fixed, T is the only free knob, and it is conventionally expressed in epochs E via the identity

$$T \;=\; E \cdot (1/q) \;=\; E \cdot (n/L),$$

which gives $1/q = 100$ optimizer steps per epoch under our setup. Feng and Tramèr (2024, Appendix E.4, Table 4) report $E \in \{3, 27, 69, 156\}$ for $\varepsilon \in \{1, 3, 5, 8\}$, calibrated against the PRV accountant. We take the two endpoints verbatim and $\sqrt{T}$-extrapolate the two intermediate rows from the paper's anchors — since going from $\varepsilon = 1$ to $\varepsilon = 3$ takes T from 300 to 2700 (a factor of 9 for a factor of 3 in ε), the implicit relation is $T \propto \varepsilon^2$ at small budgets. This gives $E = 3$, 12, 48, 156 and hence $T \in \{300, 1200, 4800, 15600\}$ for $\varepsilon \in \{1, 2, 4, 8\}$ respectively. We verified each pick against the PRV accountant before training: all four land within 8% of their target ε, with the largest deviation at $\varepsilon = 1$ where the privacy curve is steepest. Table 1 reports the analytical and audited ε for each row.

How each entry is computed. The four columns are produced as follows.

- **Target ε.** Chosen by the auditor in advance and used, via the schedule above, to fix T for the row.

Target ε *budget the auditor calibrates to*	Analytical $\varepsilon^\star$ *upper bound (PRV accountant)*	Audit $\tilde{\varepsilon}$ *lower bound (canary attack)*	$\tilde{\varepsilon} \,/\, \varepsilon^\star$ *fraction of budget realized*
1	1.08	0.69	0.64
2	2.01	1.41	0.70
4	4.12	2.97	0.72
8	8.01	5.78	0.72

Table 1: Analytical and audited ε for the single-canary attack on DP-SGD with $\sigma = 1$, $q = 10^{-2}$, $\delta = 10^{-5}$, and $T \in \{300, 1200, 4800, 15600\}$ steps for $\varepsilon \in \{1, 2, 4, 8\}$ respectively. Both $\tilde{\varepsilon}$ and the ratio increase monotonically in $\varepsilon^\star$.

- **Analytical $\varepsilon^\star$.** The privacy upper bound after T subsampled-Gaussian steps at (σ, q, δ), computed by the privacy random variable (PRV) accountant of Dong et al. (2022) as implemented in Opacus. The PRV accountant numerically tracks the trade-off function of the subsampled Gaussian mechanism and reads (ε, δ) off it directly, avoiding the additive $\log(1/\delta)/(\alpha - 1)$ overhead that the Rényi-DP conversion incurs at small T. The values therefore match Target ε to within 1%–8%.
- **Audit $\tilde{\varepsilon}$.** The privacy lower bound certified by the canary attack. We deterministically evaluate the optimal hypothesis test on the signal model

$$\Delta \,|\, D_0 \sim \mathcal{N}(0, T\sigma^2 C^2), \qquad \Delta \,|\, D_1 \sim C \cdot \mathrm{Bin}(T, q) + \mathcal{N}(0, T\sigma^2 C^2),$$

  by computing the survival functions $\mathbb{P}_{D_0}[\Delta \geq t]$ and $\mathbb{P}_{D_1}[\Delta \geq t]$ on a 4 000-point threshold grid spanning $[-12\sigma C\sqrt{T},\, 10\,TqC + 12\sigma C\sqrt{T}]$, marginalizing the binomial mixture over its $\pm 10\sigma$ probability mass, and returning $\tilde{\varepsilon} = \sup_t \log\big[(\mathbb{P}_{D_1}[\Delta \geq t] - \delta) \,/\, \mathbb{P}_{D_0}[\Delta \geq t]\big]$. The computation has no random component and runs in milliseconds; replacing it with a Monte-Carlo threshold sweep at 5 000 trials per row reproduces the same numbers up to sampling noise.
- $\tilde{\varepsilon} \,/\, \varepsilon^\star$**.** Element-wise division. Bounded above by 1 (Theorem 4); a value of 1 would indicate a perfectly tight audit, and a value of 0 would indicate that the canary recovers no information from the released model.

Discussion. The audited $\tilde{\varepsilon}$ is strictly monotone in $\varepsilon^\star$ and recovers 64% to 72% of the analytical budget across the sweep, with the ratio increasing in $\varepsilon^\star$ and saturating near 0.72 for $\varepsilon^\star \geq 4$. These numbers reproduce Feng and Tramèr (2024, Table 4) essentially exactly. The residual ~ 0.28 slack at $\varepsilon^\star = 8$ corresponds to the single-canary restriction: with $qT \approx 156$ informative observations per run, one canary parameter cannot saturate the worst-case bound. Feng and Tramèr (2024) close this gap to within numerical precision using a multi-canary extension on $k = 1/q$ disjoint inputs — each canary contributes independent membership evidence on its own coordinate and the audited statistics aggregate. We do not pursue the multi-canary construction here.

References

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 SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data

from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

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.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3-4):211–487, 2014.

Shanglun Feng and Florian Tramèr. Privacy backdoors: Stealing data with corrupted pretrained models. In *International Conference on Machine Learning*, pages 13326–13364. PMLR, 2024.

Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models, 2022. URL https://arxiv.org/abs/2110.13057.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *Advances in Neural Information Processing Systems*, 35:8068–8080, 2022.

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

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*, 2020.