Differentially Private Model Merging

Qichuan Yin*
The University of Chicago
qichuan@uchicago.edu

Manzil Zaheer
Google DeepMind
manzilzaheer@google.com

Tian Li
The University of Chicago
litian@uchicago.edu

Abstract

In machine learning, privacy requirements at inference or deployment time often evolve due to changing policies, regulations, or user preferences. In this work, we aim to construct a magnitude of models to satisfy any target differential privacy (DP) requirement *without additional training*, given a set of existing models trained on the same dataset with different privacy/utility tradeoffs. We propose two post-processing techniques, namely random selection and linear combination, to generate final private models satisfying any target privacy parameter. We provide privacy accounting of these approaches from the lens of Rényi DP and privacy loss distributions on general problems, as well as on private mean estimation, where we precisely characterize the privacy/utility tradeoffs and compare the two mechanisms. Empirically, we demonstrate the effectiveness of our approaches and validate our analyses on several models and both synthetic and real-world datasets.

1 Introduction

Differential privacy (DP) [6] offers a rigorous statistical framework to quantify privacy leakage of random algorithms and has been widely used in practice [e.g., 2]. During inference or deployment time of real-world applications, the privacy requirements can constantly change due to evolving regulatory and policy contexts, or dynamic user experience [3]. It is therefore critical to update the deployed models quickly while satisfying the target privacy constraints. In the presence of new target privacy constraints, re-training (or finetuning) a model with DP may need extra hyperparameter tuning (e.g., noise multiplier and gradient clipping threshold) and its implementation of per-gradient clipping can incur significant computation and systems costs [19].

On the other hand, it is not uncommon for service providers to have access to multiple DP models (e.g., trained with different hyperparameters) with varying privacy parameters. For instance, during model development stage, one may train a non-private model as a reference, and train a portfolio of private models (on the same dataset) to explore the privacy/utility tradeoff. In this work, we therefore ask: *can we merge the existing models with different privacy/utility tradeoffs to output a magnitude of models that satisfy certain target privacy requirement efficiently without touching raw data or performing any additional training?*

Though various variants of model merging have been extensively studied in prior literature [e.g., 23], it remains open how to merge private models given a target privacy, and how to perform tight accounting of privacy guarantees. For instance, existing privacy composition results (e.g., sequential or advanced composition [7]) are not tailored to any model merging algorithms and may result in a suboptimal privacy/utility tradeoff. To this end, we propose two simple, data-independent algorithms that enable modular and efficient combination of multiple private models to satisfy changing privacy requirements during inference time. We further provide tailored privacy accounting based on Rényi differential privacy [18] and privacy loss distributions [17, 21] for toy and general ML problems.

*Corresponding author: qichuan@uchicago.edu

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

The first merging algorithm is based on the intuitive idea of randomly outputting any model based on some probability distribution (named random selection, or **RS**). And the second algorithm is to perform a linear combination of the models (named linear combination, or **LC**). That is to say, suppose the algorithm takes as input two models θ_1 and θ_2,[2] RS flips a coin independent of the data and outputs θ_1 with probability π, and θ_2 otherwise. LC outputs the deterministic result $\theta_\lambda = \lambda\theta_1 + (1-\lambda)\theta_2$. In both cases, we provide algorithms to choose mixing coefficients π and λ, aiming to optimize utilities subject to the privacy constraint. Our contributions can be summarized as below.

- We study the problem of merging private models to satisfy dynamic privacy requirements during deployment. We introduce two lightweight post-processing mechanisms, random selection (RS) and linear combination (LC), as well as algorithms for setting merging parameters.
- We provide privacy accounting for both mechanisms based on Rényi differential privacy and privacy loss distributions on general problems. We also examine a case study on a mean estimation problem where linear combination dominates random selection.
- We validate our theoretical claims via empirical evaluation. Additionally, the experiments demonstrate that both RS and LC improve privacy/utility tradeoffs than those under naive privacy accounting in both synthetic and real-world datasets.

2 Related Work and Preliminaries

Model Merging. Model merging has emerged as a widely used technique for combining multiple trained models, but focuses on non-private settings. In our paper, model merging refers to merging multiple trained models in parameter space. Existing model merging methods can be broadly categorized into: (i) simple averaging methods such as model soups [23], which directly average model parameters with data-independent weights; our LC method is closest to this family; (ii) data-dependent weighted averaging methods such as Fisher-weighted averaging [16], which are similar to LC but typically require access to data to determine the weights, making privacy accounting more complex; (iii) parameter-dependent merging methods such as TIES [24], which determine the merge at the parameter level based on properties of the parameters themselves (e.g., sign agreement); these methods often involve nonlinear operations, which make the merged model harder to analyze from a privacy perspective; and (iv) checkpoint merging methods such as SWA [11].

Closest to our setting, recent work on merging in private learning mainly studies checkpoint or model aggregation as post-processing tools to improve generalization under DP-SGD noise. This direction is particularly appealing because DP-SGD introduces additional noise and often yields less stable checkpoints. Indri et al. [10] adapt stochastic weight averaging [11] to the DP setting, and find that averaging checkpoints can improve performance and reduce variance. Shejwalkar et al. [20] study broader aggregations of intermediate checkpoints. However, most existing work primarily targets utility improvement rather than flexible privacy adaptation, and DP-SGD checkpoint merging generally cannot guarantee privacy beyond the least private checkpoint, as discussed in Appendix D.3. To the best of our knowledge, no prior work systematically studies merging private models under target DP constraints.

Differential Privacy. Differential privacy (DP) (Definition 2.1) is a classic statistical framework of measuring privacy properties for randomized algorithms [6, 8, 1, 3] . In machine learning, the dominant approach to enforcing DP constraints is to use DP variants of stochastic gradient descent (DP-SGD [1]), which clip per-example gradients and add calibrated noise at each iteration.

Definition 2.1 (Differential Privacy [6])**.** A randomized mechanism $\mathcal{M}$ that operates on datasets consisting of individual datapoints is (ε, δ)-*DP* if for all (measurable) subsets S of outputs and for all neighboring datasets D, D' that have the same size and differ in exactly one record, we have $\Pr[\mathcal{M}(D) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(D') \in S] + \delta$. When $\delta = 0$, we simply say that $\mathcal{M}$ is ε-*DP*.

Popular privacy accounting techniques, such as those based on Rényi differential privacy (RDP) and privacy loss distributions (PLD) [18, 22, 4, 17, 9, 13, 5, 21, 25], yield tighter bounds on the cumulative privacy loss and have become standard in practical deployments of DP-SGD. In this work, we explore RDP and PLD accountants to audit privacy of our proposed approaches. For neighboring

[2] Our framework handles N models in the rest of the paper.

datasets D and D', RDP uses Rényi divergence of order α (denoted as D_α) to measure the difference between output distributions, forming a curve (*RDP parameters* or *profiles*) $\alpha \mapsto \varepsilon_\alpha$, where

$$\varepsilon_\alpha \;=\; \sup_{D \sim D'} \mathcal{D}_\alpha\big(\mathcal{M}(D) \,\|\, \mathcal{M}(D')\big). \tag{1}$$

The main benefit is that it composes additively across T steps: $\varepsilon_\alpha^{\text{total}} = \sum_{t=1}^{T} \varepsilon_\alpha^{(t)}$. We can then convert $(\alpha, \varepsilon_\alpha)$-RDP into standard (ε, δ)-DP [18]. PLD-based accounting instead works with the hockey-stick divergence

$$\mathcal{H}_\varepsilon(P, Q) \;\triangleq\; \sup_S \big\{P(S) - e^\varepsilon Q(S)\big\}. \tag{2}$$

Here P and Q denote the (pessimistic estimate of) output distributions of a mechanism $\mathcal{M}$ on neighboring datasets D and D', and S ranges over all measurable events. We can track the ***PLD parameters*** or *profiles* via $\delta(\varepsilon) = \max\{\mathcal{H}_\varepsilon(P, Q), \mathcal{H}_\varepsilon(Q, P)\}$, and compose privacy loss over steps via convolution [21]. Although PLD often gives tighter numerical bounds, we also report RDP results because RDP yields simpler analytical expressions, is useful for theoretical comparisons (e.g., Section 4.1), and is computationally lightweight and widely supported in DP pipelines.

3 Problem Formulation

We consider N differentially private algorithms $\mathcal{M}_1, \ldots, \mathcal{M}_N$ applied on the same dataset D and each outputting parameters $\theta_1(D), \ldots, \theta_N(D) \in \mathbb{R}^p$. For $i \in \{1, \ldots, N\}$, the training mechanism $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i)$-DP (Definition 2.1). Here we assume all input models must be private ($\varepsilon_i < \infty$) unless $\varepsilon' \to \infty$. Our goal is to construct a (possibly randomized) mechanism h that takes as input these N models along with the algorithms, and outputs

$$\theta(D) \;=\; h\big(\{\theta_i(D)\}_{i=1}^N, \{\mathcal{M}_i\}_{i=1}^N; \varepsilon', \delta'\big)$$

that satisfies target (ε', δ')-DP. Obviously, for suitably large (ε', δ'), there always exists a trivial solution set $h\big(\{\theta_i(D)\}_{i=1}^N, \{\mathcal{M}_i\}_{i=1}^N; \varepsilon', \delta'\big) \in \{h : h \text{ depends only on the most private } \theta_i(D)\}$. Therefore, rather than identifying a single construction, we aim to characterize a large class of feasible merged outputs with as tight privacy accounting as possible, which translates to better privacy/utility tradeoffs.

Necessity of Knowing $\{\mathcal{M}_i\}_{i=1}^N$. Ideally, we would like the merging algorithm h to depend only on the N model parameters $\{\theta_i(D)\}_{i=1}^N$ without having access to the hyperparameters of the training algorithms $\{\mathcal{M}_i\}_{i=1}^N$ that produce them. However, such mechanism-agnostic post-processing merging algorithms may be fundamentally limited. In particular, next we show that when $N = 2$, under mild regularity conditions, any h that outputs an (ε', δ')-DP model must lie very close to the class $\{h : h \text{ depends only on the more private } \theta_i(D)\}$.

Proposition 3.1 (No-free-lunch without knowing $\{\mathcal{M}_i\}_{i=1}^N$)**.** *Assume the mapping $h = h(\theta_1(D), \theta_2(D); \varepsilon', \delta')$ (either deterministic or random) is measurable and independent of $(\mathcal{M}_1, \mathcal{M}_2)$. Assume $\varepsilon_1 < \varepsilon' < \varepsilon_2$ and $\delta_1 = \delta_2 = \delta$. If*

$$\sup_{x,a,b \in \mathbb{R}^p} \mathcal{D}_{TV}(h(x, a), h(x, b)) > \frac{(\delta' + e^{\varepsilon'} - 1)(e^{\varepsilon_2} + 1)}{e^{\varepsilon_2 + \varepsilon'} + \delta e^{\varepsilon'} + \delta - 1}, \tag{3}$$

then the output $h\big(\theta_1(D), \theta_2(D)\big)$ cannot be (ε', δ')-DP for all admissible choices of $(\mathcal{M}_1, \mathcal{M}_2)$.

The complete proof is provided in Appendix C.1. Condition 3 describes mergers $h(\cdot, \cdot)$ that react too much to changes in the less private model θ_2. If varying θ_2 while fixing θ_1 can induce a large total-variation change in the distribution of $h(\theta_1, \theta_2)$, then no universal privacy guarantee is possible. Proposition 3.1 formalizes that, without structural assumptions on $\mathcal{M}_1, \mathcal{M}_2$, one cannot design a merging algorithm that is substantially different from the trivial merger while satisfying (ε', δ')-DP. In particular, when $\delta' = \delta$, any deterministic merger that aims to be valid uniformly over $(\mathcal{M}_1, \mathcal{M}_2)$ must lie in the trivial class.

Therefore, to obtain meaningful improvements beyond the trivial merger, we focus on practically relevant mechanisms rather than arbitrary ones. In particular, we study the setting where the inputs are associated with some known RDP or PLD parameters (which can be inferred from hyperparameters

of the learning algorithms) (Section 5), or where the inputs are directly produced by running DP-SGD (Section 6). In the next sections, we first formally introduce two model merging algorithms (two instances of h), random selection and linear combination (Section 4). We then discuss privacy accounting of RS (Section 5) and LC (Section 6) mechanisms, and consider how to choose an appropriate merger h that exploits the additional assumptions of $\mathcal{M}$.

4 Model Merging via Random Selection or Linear Combination

In this section, we first formally propose two data-independent merging mechanisms, and illustrate the privacy/utility tradeoff introduced by these mechanisms on a Gaussian mean estimation problem (Section 4.1).

Random Selection (RS). We propose a mechanism independent of D that samples a model index I from a categorical distribution and outputs $\theta_I(D)$ as the merged model. That is, given target privacy (ε', δ'), we output

$$\theta(D) = \theta_I(D), \quad I \sim \mathrm{Cat}(\pi_1, \ldots, \pi_N), \tag{4}$$

where $\pi = (\pi_1, \ldots, \pi_N) \in \Delta^{N-1}$ (a probability simplex) and π is dependent on (ε', δ') (Section 5). Note that RS only randomly samples one model following some distribution. Another natural choice is to leverage all the models by linearly averaging them.

Linear Combination (LC). The LC method for N models on dataset D is defined as

$$\theta(D) \;=\; \sum_{i=1}^{N} \lambda_i \, \theta_i(D), \; \lambda = [\lambda_1, \cdots, \lambda_N] \in \Delta^{N-1}. \tag{5}$$

In both cases, as h does not have access to D, the privacy of composing N models is fully determined by how the privacy losses of $\{\mathcal{M}_i\}_{i=1}^N$ are aggregated under different instantiations of h, and the main task becomes privacy accounting for the RS or LC procedure.

At a high level, RS is entirely characterized by privacy parameters of RDP or PLD, as the output model is randomly sampled from the input N models. Therefore, its privacy accounting depends on the individual selected model's privacy parameters, making RS broadly applicable even when we do not understand the detailed procedure of the training algorithm (as long as we have access to privacy parameters). In contrast, LC requires to imposing stronger assumptions on the structures (e.g., hyperparameters and updating rules) of the training algorithms that produce private input models (formalized in Section 6). Next, we study a toy problem on a Gaussian mean estimation to illustrate this intuition, before introducing general accounting.

4.1 Case Study: Mean Estimation

This section studies Gaussian mean estimation, one of the simplest statistical tasks where we can precisely characterize the privacy and utilities of RS and LC, under RDP accounting. In this setting, we show that LC can match the privacy of RS while achieving no larger MSE. Let $\hat{\mu}$ be the empirical mean with ℓ_2 sensitivity Δ, and consider two independent Gaussian mechanisms $\theta_i(D) \sim \mathcal{N}(\hat{\mu}, \sigma_i^2), \quad i = 1, 2$. For RS with parameter π, the output follows $\pi\mathcal{N}(\hat{\mu}, \sigma_1^2) + (1-\pi)\mathcal{N}(\hat{\mu}, \sigma_2^2)$, with variance $\sigma^2_{\mathrm{RS}}(\pi) = \pi\sigma_1^2 + (1-\pi)\sigma_2^2$. For LC with parameter λ, the output follows $\mathcal{N}\big(\hat{\mu}, \lambda^2\sigma_1^2 + (1-\lambda)^2\sigma_2^2\big)$, with variance $\sigma^2_{\mathrm{LC}}(\lambda) = \lambda^2\sigma_1^2 + (1-\lambda)^2\sigma_2^2$. Since both estimators are unbiased relative to $\hat{\mu}$, comparing their MSE reduces to comparing these variances.

Theorem 4.1. *Fix any $\alpha > 1$ and $\pi \in [0, 1]$. There exists a constant $C_{\pi,\alpha} > 0$ such that, whenever $\Delta < C_{\pi,\alpha}$, the following holds: if $\sigma^2_{\mathrm{LC}}(\lambda) = \sigma^2_{\mathrm{RS}}(\pi)$, then LC with parameter λ achieves privacy no worse than RS with parameter π under order-α RDP. Consequently, for any RS parameter π, there exists an LC parameter λ that achieves no larger MSE under the same target order-α RDP budget.*

Thus, in this Gaussian setting, LC improves over RS as it averages independent noise rather than discarding one candidate. This clean comparison relies on the additive Gaussian structure and does not directly extend to general learning problems. We next develop accounting rules for RS under general RDP/PLD profiles, and for LC under the additional structure of independent DP-SGD runs.

Algorithm 1 Setting Selection Probabilities for RS (See Algorithm 2 in the appendix for LC)

Input: target privacy (ε', δ'), orders grid $\mathcal{A}$ for RDP, hyperparameter sets $\{\mathcal{S}_i\}_{i=1}^N$ of $\{\mathcal{M}_i\}_{i=1}^N$.
Output: set of selection probabilities $\pi \in \Delta^{N-1}$ satisfying the target privacy.

Case 1: RDP accounting

Obtain $\{\varepsilon_{\alpha,i}\}_{\alpha\in\mathcal{A}}$ by $\mathcal{S}_i$, for each $i \in [N]$
function DP_EPS(π, δ)
  for $\alpha \in \mathcal{A}$ **do**
    $a_i \leftarrow (\alpha - 1)\,\varepsilon_{\alpha,i},\ i \in [N]$
    $\varepsilon_\alpha^{\mathrm{RS}}(\pi) \leftarrow \frac{\log\left(\sum_{i=1}^N \pi_i e^{a_i}\right)}{\alpha-1}$
    $\varepsilon_\alpha(\pi;\delta) \leftarrow \varepsilon_\alpha^{\mathrm{RS}}(\pi) + \frac{\log(1/\delta)}{\alpha-1}$
  end for
  return $\min_{\alpha\in\mathcal{A}} \varepsilon_\alpha(\pi;\delta)$
end function
return $\{\pi \mid \text{DP_Eps}(\pi, \delta') \le \varepsilon'\}$

Case 2: PLD accounting

Obtain $\varepsilon \mapsto \{\mathcal{H}_\varepsilon(P_i, Q_i),\ \mathcal{H}_\varepsilon(Q_i, P_i)\}$ by $\mathcal{S}_i$, for each $i \in [N]$.
function DP_DELTA(π, ε)
  $\delta_+ \leftarrow \sum_{i=1}^N \pi_i\, \mathcal{H}_\varepsilon(P_i, Q_i)$
  $\delta_- \leftarrow \sum_{i=1}^N \pi_i\, \mathcal{H}_\varepsilon(Q_i, P_i)$
  $\delta^{\mathrm{RS}}(\pi;\varepsilon) \leftarrow \max\{\delta_+, \delta_-\}$
  return $\delta^{\mathrm{RS}}(\pi;\varepsilon)$
end function
return $\{\pi \mid \text{DP_Delta}(\pi, \varepsilon') \le \delta'\}$

5 Privacy of Random Selection

Recall that we have trained N private models on the same dataset D, obtaining output parameters $\theta_1(D), \ldots, \theta_N(D) \in \mathbb{R}^p$. As discussed in Section 3, there is no general post-processing algorithm that improves utilities beyond the degenerated solutions if we do not have access to any parameters of the training algorithms $\{\mathcal{M}_i\}_{i=1}^N$. Hence, for random selection, we assume that we are given the hyperparameters of $\mathcal{M}$, from which we derive the RDP and PLD profiles of each input model. For each $\mathcal{M}_i$, we denote the RDP parameters as $\{\varepsilon_{\alpha,i}\}_{\alpha>1}$ and the PLD privacy parameters as $\varepsilon \mapsto \delta_i(\varepsilon)$. To this end, we develop two complementary accounting techniques: an RDP-based bound that is convenient for analysis and optimization, and a PLD-based method that is typically tighter in practice.

RDP Accounting. We begin by establishing an explicit upper bound on the RDP parameters of RS.

Theorem 5.1. *For any $\alpha > 1$, random selection (Eq. (4)) over N private models (each being $(\alpha, \varepsilon_{\alpha,i})$-RDP) satisfy $(\alpha, \varepsilon_\alpha^{RS}(\pi))$-RDP where $\varepsilon_\alpha^{RS}(\pi) \le \frac{1}{\alpha-1} \log\Big(\sum_{i=1}^N \pi_i\, e^{(\alpha-1)\varepsilon_{\alpha,i}}\Big)$.*

Theorem 5.1 (proved in Appendix C.3) provides an explicit upper bound on the RDP parameter of random selection in terms of the individual model's parameters $\{\varepsilon_{\alpha,i}\}_{i=1}^N$ and the mixing weights $\pi \in \Delta^{N-1}$. This bound immediately enables us to certify $(\varepsilon(\pi;\delta'), \delta')$-DP for candidate mixtures via the standard RDP-to-DP conversion. For a target privacy budget ε', we accept a mixing weight π if $\varepsilon(\pi;\delta') \le \varepsilon'$. This provides us the criterion for whether a candidate π satisfies the target privacy guarantee. We summarize the method in Algorithm 1 (left).

PLD Accounting. In the PLD accounting, we note that random selection induces the mixture distributions over the N models. That is, considering any neighboring datasets D and D', the output probabilities of RS are

$$P_{RS}^D = \sum_{i=1}^N \pi_i P_i^D, \quad P_i^D = \mathrm{Law}(\mathcal{M}_i(D)), \quad Q_{RS}^{D'} = \sum_{i=1}^N \pi_i Q_i^{D'}, \quad Q_i^{D'} = \mathrm{Law}(\mathcal{M}_i(D'))$$

where $\mathrm{Law}(X)$ refers to the distribution induced by random variable X. Further, we define P_i and Q_i to be pessimistic estimates of P_i^D and $Q_i^{D'}$ respectively, which means $\mathcal{H}_\varepsilon(P_i^D, Q_i^{D'}) \le \mathcal{H}_\varepsilon(P_i, Q_i)$ and $\mathcal{H}_\varepsilon(Q_i^{D'}, P_i^D) \le \mathcal{H}_\varepsilon(Q_i, P_i)$. Note that the hockey-stick divergence used by PLD accounting $\mathcal{H}_\varepsilon(\cdot,\cdot)$ (Eq. (2)) is jointly convex over (P, Q). Then we have the following result about the PLD parameter $\delta^{\mathrm{RS}}(\varepsilon)$ of the random selection output.

Theorem 5.2. *For any $\varepsilon \ge 0$ and $\pi = (\pi_1, \ldots, \pi_N) \in \Delta^{N-1}$, it holds that*

$$\sup_{D\sim D'} \mathcal{H}_\varepsilon(P_{RS}^D, Q_{RS}^{D'}) \le \sum_{i=1}^N \pi_i\, \mathcal{H}_\varepsilon(P_i, Q_i), \quad \sup_{D\sim D'} \mathcal{H}_\varepsilon(Q_{RS}^{D'}, P_{RS}^D) \le \sum_{i=1}^N \pi_i\, \mathcal{H}_\varepsilon(Q_i, P_i).$$

Consequently, we have $\delta^{RS}(\varepsilon)$ is upper bounded by $\max\Big\{\sum_{i=1}^N \pi_i \mathcal{H}_\varepsilon(P_i, Q_i), \sum_{i=1}^N \pi_i \mathcal{H}_\varepsilon(Q_i, P_i)\Big\}$.

Theorem 5.2 (proved in Appendix C.4) also yields a simple feasibility test for RS based on the privacy loss distribution. Given a target privacy parameter ε', we evaluate $\delta^{\mathrm{RS}}(\varepsilon')$ for a candidate probability π and π is feasible if and only if $\delta^{\mathrm{RS}}(\varepsilon') \le \delta'$. We summarize this method in Algorithm 1 (right).

Privacy/Utility Tradeoff under Random Selection. For a target (ε', δ'), the accounting methods above provide a feasible set of $\pi \in \Delta^{N-1}$. In practice, one would naturally like to choose π that yields a strong utility among all feasible π's. This calls for selection rules over π. Under random selection, the expected utility is linear in π, making the tradeoff easier to reason about. In typical DP scenarios, more private models tend to have lower utilities, so a natural strategy is to choose π that nearly saturates the privacy budget, e.g., $\varepsilon(\pi; \delta') \approx \varepsilon'$ in RDP, thereby biasing selection towards higher-utility models while maintaining the privacy guarantee. Moreover, many selection rules are possible in practice. For example, one may evaluate candidate mixtures on a public dataset and choose the best-performing option. We will discuss selection on a private data in Section 6.3.

6 Privacy of Linear Combination

In Section 4.1, we leverage the additivity of independent Gaussian noise and show that, for mean estimation with a Gaussian mechanism, LC strictly dominates RS. However, such favorable structure does not hold for general problems. For random selection, the only assumption we make is access to the RDP/PLD parameters of each input model. In contrast, we show in this section that this is not sufficient for linear combination. That is, if we only know the privacy parameters and impose no further assumptions, then the privacy under LC cannot improve upon that of simply releasing all the models $\{\theta_1, \ldots, \theta_N\}$ jointly. This observation (formalized in Section 6.1 below) motivates the need for additional structural assumptions to obtain meaningful privacy accounting for LC (Section 6.2).

6.1 Limitation Result without Knowing Original Training Algorithms

Suppose we *only* know the final RDP or PLD privacy parameters of each private learning mechanism $\{\mathcal{M}_i\}_{i=1}^N$ without any further algorithmic structure information (i.e., the exact updating rules) about $\{\mathcal{M}_i\}_{i=1}^N$. We show below, when the mechanisms are instantiated with fresh independent randomness conditioned on the dataset D, the LC mechanism admits a universal worst-case upper bound, and this bound can be achieved.

Theorem 6.1 (Worst-case bounds and tightness)**.** *Consider the linear combination mechanism defined by $\theta(D) = \sum_{i=1}^N \lambda_i \theta_i(D)$. Assume $\lambda_i > 0$ and the inputs $\{\theta_1(D), \cdots, \theta_N(D)\}$ are generated with fresh independent randomness conditioned on the dataset D. We have the following results.*

(1) RDP upper bound. *The RDP parameter of θ is no worse than that of the joint release (i.e., releasing $\{\theta_1, \cdots, \theta_N\}$). Thus, for every $\alpha > 1$, the LC mechanism satisfies $(\alpha, \varepsilon_\alpha^{LC}(\lambda))$-RDP where*

$$\varepsilon_\alpha^{LC}(\lambda) \;\le\; \sum_{i=1}^N \varepsilon_{\alpha,i}. \tag{6}$$

(2) PLD upper bound. *Let ω_i be the PLD of θ_i, and let $\omega_{\mathrm{joint}} = \omega_1 \star \cdots \star \omega_N$ be the PLD of the joint release. Then the PLD parameter of θ is no worse than that of the joint release. That is, for any $\varepsilon \ge 0$,*

$$\delta^{LC}(\varepsilon) \;\le\; \delta_{\mathrm{joint}}(\varepsilon), \tag{7}$$

where $\delta_{\mathrm{joint}}(\varepsilon)$ is computed from ω_{joint}.

(3) Tightness. *The bounds in Eq.* (6)-(7) *are worst-case tight: there exist mechanisms $\{\mathcal{M}_i\}_{i=1}^N$ under which the equality holds.*

Theorem 6.1 (proved in Appendix C.5) establishes that the privacy parameter of the deterministic linear combination is fully controlled by that of the joint model, without additional assumptions. Moreover, the bound is worst-case tight; so LC satisfies no better privacy guarantee gain beyond the joint release $\{\theta_1, \cdots, \theta_N\}$. Therefore, without additional information beyond the input models' privacy parameters, one cannot derive a universally tighter privacy guarantee for LC. In particular, the resulting joint-release bounds are substantially larger than the RS bounds in Theorems 5.1 and 5.2, which are always upper-bounded by the maximum RDP/PLD parameter among all input models. This highlights that, without further structure, LC is not guaranteed to outperform RS.

6.2 Linear Combination under DP-SGD

Here, we assume that the input models are trained independently via DP-SGD [1], one of the most popular DP algorithms in ML. For clarity and ease of analysis, we consider the full-batch version.

At step $t = 1, \dots, T$, each mechanism $i \in [N]$ computes a clipped gradient $\tilde{g}_i^{(t)} = \mathrm{clip}\big(g_i^{(t)}, C_i^{(t)}\big)$ such that $\|\tilde{g}_i^{(t)}\|_2 \le C_i^{(t)}$, and add Gaussian noise $e_i^{(t)} \sim \mathcal{N}\Big(0, \big(\sigma_i^{(t)} C_i^{(t)}\big)^2 I\Big)$, independently across i. The local update is

$$\theta_i^{(t)} = \theta_i^{(t-1)} - \eta_i^{(t)}\big(\tilde{g}_i^{(t)} + e_i^{(t)}\big).$$

Define the model output via LC after step t as $\theta^{(t)} = \sum_{i=1}^N \lambda_i \theta_i^{(t)}$. The combined update can therefore be written as

$$\theta^{(t)} - \theta^{(t-1)} \;=\; -\sum_{i=1}^N \lambda_i \, \eta_i^{(t)}\big(\tilde{g}_i^{(t)} + e_i^{(t)}\big) \;:=\; -\big(\overline{g}_\lambda^{(t)} + \overline{e}_\lambda^{(t)}\big),$$

where $\overline{g}_\lambda^{(t)} := \sum_{i=1}^N \lambda_i \eta_i^{(t)} \tilde{g}_i^{(t)}$ and $\overline{e}_\lambda^{(t)} := \sum_{i=1}^N \lambda_i \eta_i^{(t)} e_i^{(t)}$. Note that the aggregated noise $\overline{e}_\lambda^{(t)}$ follows the distribution $N\big(0, (s_\lambda^{(t)})^2 I\big)$, where $(s_\lambda^{(t)})^2 := \sum_{i=1}^N \big(\lambda_i \, \eta_i^{(t)} \, \sigma_i^{(t)} \, C_i^{(t)}\big)^2$. Define the mixed sensitivity $\Delta_\lambda^{(t)} = \sum_{i=1}^N \lambda_i \eta_i^{(t)} \Delta_i^{(t)}$, where $\Delta_i^{(t)}$ is the sensitivity of mechanism i for step t.

A subtle point is that standard RDP and PLD composition (shown in Appendix A.1) is applied by conditioning on the previously accounted output. In our LC analysis, this means conditioning on the previous merged parameter $\theta^{(t-1)}$, rather than on the individual parameter $\{\theta_i^{(t-1)}\}_{i=1}^N$. Consequently, given dataset D, the individual gradients $\widetilde{g}_i^{(t)}$ are not fixed after conditioning on $\theta^{(t-1)}$; the conditional distribution of the combined update $\theta^{(t)} - \theta^{(t-1)}$ is generally a mixture over all possible values of $\{\theta_i^{(t-1)}\}_{i=1}^N$, rather than a single Gaussian.

Let $\mathcal{F}_{t-1}$ denote the sigma field generated by $\theta^{(t-1)}$. The following results show that, although the conditional distribution of $\{\theta_i^{(t-1)}\}_{i=1}^N$ is unknown and the resulting combined update is generally non-Gaussian, valid RDP and PLD accounting bounds for LC can still be obtained.

Theorem 6.2. *For any order $\alpha > 1$, conditional on $\mathcal{F}_{t-1}$, the RDP bound at step-t for the LC mechanism satisfies:*

$$\varepsilon_\alpha^{(t)}(\lambda) \le \alpha(\Delta_\lambda^{(t)})^2 / \left(2(s_\lambda^{(t)})^2\right). \tag{8}$$

Full proof of Theorem 6.2 is in Appendix C.6. In practice, we apply Theorem 6.2 by summing the per-step privacy costs over all T iterations using the standard RDP composition rule in Appendix A.1, and then converting the resulting RDP parameters into DP guarantees. We summarize the accounting procedures in Algorithm 2 (left) in the appendix. For PLD accounting, we have the following results.

Theorem 6.3. *Let $P^{(t)}$ and $Q^{(t)}$ denote the distribution of $\theta^{(t)} - \theta^{(t-1)}$ under D and D' conditional on $\mathcal{F}_{t-1}$. Consider the practical surrogate privacy loss random variable*

$$\tilde{L}^{(t)} = \log \frac{\phi\Big(Z^{(t)}; \Delta_\lambda^{(t)}, (s_\lambda^{(t)})^2\Big)}{\phi\Big(Z^{(t)}; 0, (s_\lambda^{(t)})^2\Big)}, \qquad Z^{(t)} \sim \mathcal{N}(0, (s_\lambda^{(t)})^2), \tag{9}$$

where $\phi(\cdot; \mu, \sigma^2)$ denotes Gaussian density with mean μ and variance σ^2. Then, for every $\varepsilon \ge 0$,

$$\mathcal{H}_\varepsilon(P^{(t)}, Q^{(t)}) \le \mathbb{E}[(e^{\tilde{L}^{(t)}} - e^\varepsilon)_+], \quad \mathcal{H}_\varepsilon(Q^{(t)}, P^{(t)}) \le \mathbb{E}[(1 - e^{\varepsilon + \tilde{L}^{(t)}})_+].$$

Corollary 6.4. *Under the notation of Theorem 6.3, assume $L^{(t)}$ is the real privacy loss random variable. Define $L := \sum_{t=1}^T L^{(t)}$, $\tilde{L} := \sum_{t=1}^T \tilde{L}^{(t)}$. Then, for every $\varepsilon \ge 0$, it holds that*

$$\mathbb{E}\Big[\big(e^L - e^\varepsilon\big)_+\Big] \;\le\; \mathbb{E}\Big[\big(e^{\tilde{L}} - e^\varepsilon\big)_+\Big], \quad \mathbb{E}\Big[\big(1 - e^{\varepsilon + L}\big)_+\Big] \;\le\; \mathbb{E}\Big[\big(1 - e^{\varepsilon + \tilde{L}}\big)_+\Big]$$

As discussed above, the conditional distribution of $\theta^{(t)} - \theta^{(t-1)}$ is generally non-Gaussian, so the actual privacy-loss random variable $L^{(t)}$ does not admit a simple form. The surrogate variable $\widetilde{L}^{(t)}$ in Theorem 6.3 gives a tractable Gaussian upper bound for each step. Together with Corollary 6.4 and the standard PLD composition rule in Appendix A.1, this yields a practical LC accountant by

convolving the per-step surrogate PLDs from Eq. (9). Algorithm 2 (right) summarizes the procedure; complete proofs are in Appendix C.7.

While LC does not enjoy the same clean utility/privacy tradeoff structure as RS in general, especially for nonconvex models, many practical selection rules remain applicable. In particular, LC can still be tuned effectively by evaluating performance on a public benchmark or a non-sensitive validation set. Further, we will discuss tuning on private data in Section 6.3.

Different Training Steps Across Input Models. In some applications, the input candidate models may be trained with differential privacy for different numbers of iterations $\{T_i\}_{i=1}^N$. To make the accounting go through, we can align all runs to a common iteration $T \triangleq \max_i T_i$. For any model i with $T_i < T$, we could use an equivalent T-step representation by appending $T - T_i$ virtual steps.

6.3 Selecting Merging Weights in Practice

The accounting above returns a feasible set of weights (π or λ) for a target (ε', δ'), and a rule is needed to pick one. As we mention in Section 5 and 6, without any data access, one can select a feasible weight that saturates the privacy budget, breaking ties uniformly at random; With a public or non-sensitive validation set, candidate weights can also be scored directly; when no such set is available, synthetic data sometimes can serve as a proxy, as it approximates human data reasonably well when used in moderation [26]. When only private data is available, we still could tune the weights but need additional accounting.

Proposition 6.5. *Fix a data-independent grid Λ of K feasible weights, so that for every $\lambda \in \Lambda$ the merged model θ_λ is (ε', δ')-DP, and let $s = (s_\lambda)_{\lambda\in\Lambda} \in [0,1]^K$ be any score that may depend on both the input models and the private data D. For any $t > 0$, sampling $\lambda^* \in \Lambda$ with probability $\rho_\lambda \propto e^{ts_\lambda}$ and releasing θ_{λ^*} is $(\varepsilon' + 2t,\ e^t\delta')$-DP.*

The proof is in Appendix C.9. The additional privacy cost of tuning on private data is thus a fixed additive $2t$ in ε and a factor e^t in δ, independent of the number of candidates K and of how the score is computed. s_λ can be any utility metric of θ_λ (e.g., its loss or accuracy) normalized to $[0, 1]$ and evaluated on the private data, with no sensitivity analysis required. The temperature t controls this cost. $t \to 0$ recovers the data-independent uniform rule at no cost, while larger t concentrates on high-scoring weights at a cost of $2t$. Accordingly, to meet an overall target $(\varepsilon^\star, \delta^\star)$ after tuning, the feasible grid Λ should be built at the tighter level $(\varepsilon^\star - 2t,\ e^{-t}\delta^\star)$. The statement applies verbatim to RS with λ replaced by π.

7 Comparison with Related DP Merging and Composition Methods

Recycling Checkpoints. Recent works [20, 10] improve private learning by aggregating checkpoints from a single DP-SGD run. This algorithm differs fundamentally from ours because the candidate models share randomness and early optimization steps, thus affecting accounting. Our RS accounting only requires the individual RDP/PLD profiles of each candidate model, and therefore remains applicable to recycled checkpoints; see Appendix D.3 for empirical results. In contrast, our LC accounting relies on additional structure, in particular independent DP-SGD noise across input models. These assumptions fail for correlated checkpoints, so our LC guarantees do not apply in this setting. More specifically, LC over recycled checkpoints cannot guarantee privacy better than the least private checkpoint; see Appendix D.3 for details.

Comparing with Other DP Composition. Classical DP composition bounds [e.g., 7, 12] analyze the joint release of multiple private outputs using only their $(\varepsilon_i, \delta_i)$ guarantees. In contrast, our accounting uses richer privacy profiles, such as RDP parameters across orders or PLD curves, which typically yield tighter guarantees. Moreover, our bounds exploit the structure of the merging rule itself: RS uses the mixture structure of random selection through log-sum-exp and convexity, while LC can obtain tighter bounds when the input models are produced by independent DP-SGD runs. Thus, our accounting can be sharper than generic joint-release accounting, while also characterizing how the merging weights affect privacy. A formal comparison is provided in Proposition B.1, where we show that RDP joint accounting improves over advanced composition.

8 Empirical Evaluation

In this section, we first present results on a toy mean estimation problem for RS and LC algorithms under both RDP and PLD accounting, corresponding to analysis in Section 4.1. Then on real-world datasets with both convex and non-convex models, we present the privacy/utility tradeoffs of a set of models output by RS and LC, and demonstrate tighter privacy bounds than baseline composition methods. Additional experiment details are discussed in Appendix D.

8.1 Synthetic Data: Mean Estimation

We generate $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ with $n = 100$, and then clip the samples to enforce a bounded domain, i.e., $X_j \leftarrow \text{clip}(X_j, -1, 1)$. The goal is to estimate the population mean μ for dataset $D = \{X_1, \ldots, X_n\}$. We generate two private estimates as input. Figure 1 reports the Pareto frontier between losses (mean squared error) and privacy budgets of the RS and LC algorithms under RDP and PLD accounting. PLD gives tighter privacy upper bounds than RDP. Also, both methods achieve flexible privacy by tracing out a continuous MSE/privacy tradeoff as the target privacy level changes. Moreover, LC consistently outperforms RS, validating our theoretical arguments in Section 4.1.

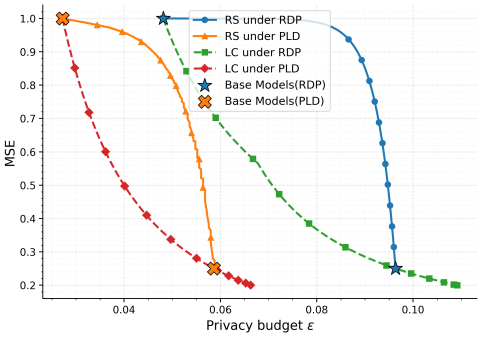
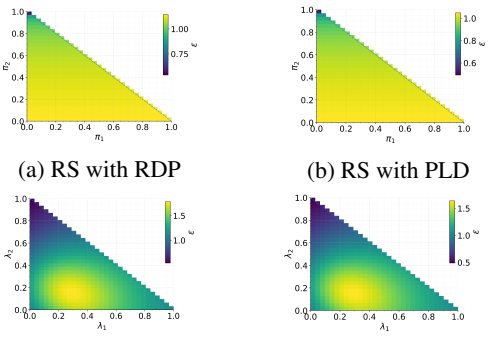


Figure 1: Privacy/utility tradeoffs of mean estimation ($\delta = 10^{-5}$). Input models are also marked in the figure.

8.2 Results on Real Datasets

We next evaluate our method on two standard benchmarks: MNIST [15] and CIFAR-10 [14], and train both convex and non-convex models. We use the standard split of MNIST for training and testing, and train a logistic regression model with DP-SGD. For CIFAR-10, we use the standard training/test split with a ResNet18 model. For both datasets, we train three input private models to cover two common sources of variation in practice. One model is obtained by changing a hyperparameter that does not affect the privacy accounting, and therefore has the same privacy guarantee but may exhibit different utility. Another model is obtained by changing a privacy-relevant hyperparameter, resulting in a different privacy/utility tradeoff. See Appendix D for hyperparameters and implementation details.



(a) RS with RDP (b) RS with PLD

(c) LC with RDP (d) LC with PLD

Figure 2: Heatmap of the privacy parameter ε on MNIST ($\delta = 10^{-5}$) as the merging weights vary when combining three models.

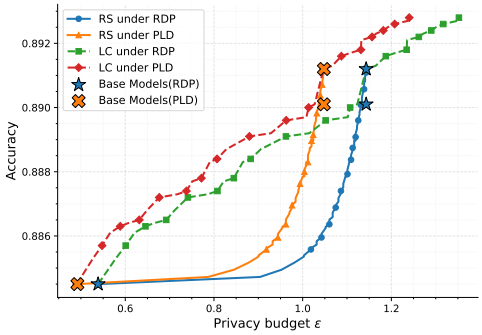


Figure 3: Privacy/utility tradeoffs of MNIST ($\delta = 10^{-5}$) on a convex model. We see that PLD is tighter than RDP in practice, and LC generally achieves a better privacy/utility tradeoff than RS.

Figure 2 reports the privacy changes as different mixing weights vary under both RS and LC with RDP and PLD. Both methods successfully adapt to flexible privacy requirements. Figure 3 shows the Pareto frontier of RS and LC with RDP and PLD on MNIST when combining three models. All methods exhibit a clear accuracy/privacy tradeoff. Across all settings, the ε values guaranteed by PLD are consistently smaller than those obtained from RDP.

We next evaluate a nonconvex setting on CIFAR-10 using a pretraining-based pipeline. We randomly split the training set into two disjoint halves, D_{pre} and D_{priv}. We first train a non-private ResNet18

model on D_{pre} using standard SGD, and then use this model as initialization for DP-SGD on D_{priv}. Starting from this initialization, we train multiple private models with different DP-SGD hyperparameters, yielding different privacy/utility tradeoffs. We then apply RS and LC to these private models as in the MNIST experiment. The privacy guarantee in this experiment is with respect to D_{priv} only, since D_{pre} is used for non-private pretraining.

Figure 4 shows the Pareto frontiers of RS and LC under both RDP- and PLD-based accounting. Both methods enable flexible adaptation to different target privacy budgets without additional training, and PLD-based accounting remains consistently tighter than RDP-based accounting. Interestingly, in this finetuning-based setting, we also observe a phenomenon that was already present in some of our earlier experiments: the merged model may outperform all individual input models. This may be because a stronger common initialization places the input models in a more well-behaved region of the parameter space, making merging more effective at preserving shared structure while averaging out part of the DP-induced noise. Similar observations have also appeared in the non-private model merging literature [23, 16]. This further highlights the value of our framework: model merging is not only a tool for adapting to different target privacy budgets after training, but may also provide utility gains relative to the input models.

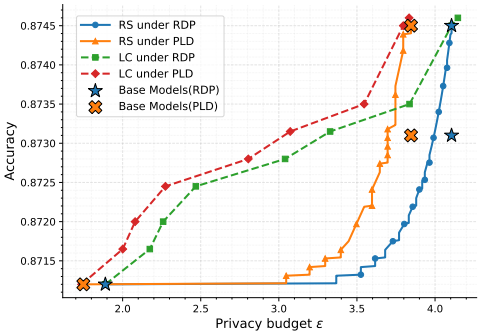


Figure 4: Privacy/utility tradeoffs of model merging on CIFAR-10 ($\delta = 10^{-5}$). Input private models are finetuned from the same pretrained model.

Additional Experiments. Appendix E reports further experiments. (i) Comparing with two natural alternatives for meeting a target budget (deploying the best eligible input model and retraining with DP-SGD at each target), both mergers are pointwise no worse than the former, and LC recovers most of the gain of retraining without any additional training. (ii) Jointly releasing all input models and then applying non-private merging methods (model soups, Fisher-weighted averaging, TIES/DARE) yields a single operating point whose joint-release privacy cost lies outside the target ranges considered here. (iii) Varying the number $N \in \{3, 5, 7\}$ of input models and the spread of their privacy levels shows that utility improves as the portfolio becomes denser, and that a wider spread trades utility for a wider range of attainable targets. (iv) Since Theorems 5.1 and 5.2 only use each candidate's RDP/PLD profile, RS also applies to input models with different architectures or initializations, whereas LC requires a common parameterization; on a heterogeneous MNIST portfolio (logistic regression, MLP, and CNN), RS still traces a privacy/utility curve while LC is not defined.

9 Conclusion, Limitation, and Future Directions

To the best of our knowledge, this is the first work that studies model merging to meet flexible privacy requirements during deployment time. We have proposed two merging strategies, based on random selection and linear combination, without any additional training steps. We provide privacy accounting based on both RDP and PLD, and our experiments demonstrate the practical accuracy/privacy tradeoffs of the proposed methods across synthetic and real datasets. Several directions remain open. One limitation of our current LC analysis is that it is developed for full-batch DP-SGD; extending the accounting to subsampled DP-SGD is an interesting direction for future work. It would also be interesting to design and analyze more sophisticated data-dependent merging algorithms that better handle challenging regimes such as nonconvex models. On the practical side, an important next step is to develop effective and private procedures for tuning merging parameters under privacy constraints. Finally, exploring the proposed technique in concrete application areas, such as human-AI collaboration and personalized agents, remains an open direction [27].

Acknowledgments and Disclosure of Funding

The work of TL was supported in part by a Google Research Award and was done in part while TL was visiting the Simons Institute for the Theory of Computing. The authors thank Marco Ciccone for valuable discussions, and the anonymous reviewers and area chair for their constructive comments. The authors declare no competing interests.

References

[1] 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.

[2] John M Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, et al. The 2020 census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, 2, 2022.

[3] Franziska Boenisch, Christopher Mühl, Roy Rinberg, Jannis Ihrig, and Adam Dziedzic. Individualized pate: Differentially private machine learning with individual privacy guarantees. *arXiv preprint arXiv:2202.10517*, 2022.

[4] 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.

[5] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *arXiv preprint arXiv:2207.04380*, 2022.

[6] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[7] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

[8] Xuchen Gong and Tian Li. Private zeroth-order optimization with public data. *Advances in Neural Information Processing Systems*, 38:58619–58665, 2025.

[9] Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

[10] Patrick Indri, Tamara Drucks, and Thomas Gärtner. Can stochastic weight averaging improve generalization in private learning? In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

[11] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[12] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[13] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 3358–3366. PMLR, 2021.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[16] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[17] Sebastian Meiser and Esfandiar Mohammadi. Tight on budget? tight bounds for r-fold approximate differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 247–264, 2018.

[18] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[19] 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, 2023.

[20] Virat Shejwalkar, Arun Ganesh, Rajiv Mathews, Yarong Mu, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Xinyi Zheng. Recycling scraps: Improving private learning by leveraging intermediate checkpoints. *arXiv preprint arXiv:2210.01864*, 2022.

[21] David Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *Cryptology ePrint Archive*, 2018.

[22] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd international conference on artificial intelligence and statistics*, pages 1226–1235. PMLR, 2019.

[23] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[24] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

[25] Xiyuan Yang, Shengyuan Hu, Soyeon Kim, and Tian Li. Differentially private federated clustering with random rebalancing. *arXiv preprint arXiv:2508.06183*, 2025.

[26] Qichuan Yin and Linwei Xin. Synthetic but not infinite: How much llm-generated data to use in market research. *Available at SSRN 6078686*, 2026.

[27] Qichuan Yin, Ziwei Su, and Shuangning Li. Overcoming the incentive collapse paradox. *arXiv preprint arXiv:2603.27049*, 2026.

[28] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

[29] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

A Composition Theory and Algorithm for LC

A.1 Composition Theory

We first recall the adaptive composition property of RDP, which justifies summing the conditional per-step RDP costs.

Theorem A.1 (Adaptive composition of RDP; Proposition 1 in [18]). *Let $f : \mathcal{D} \to \mathcal{Y}_1$ be an (α, ε_1)-RDP mechanism. Let $g : \mathcal{Y}_1 \times \mathcal{D} \to \mathcal{Y}_2$ be such that, for every $y_1 \in \mathcal{Y}_1$, the mechanism $g(y_1, \cdot)$ satisfies (α, ε_2)-RDP. Then the adaptive composition mechanism*

$$h(D) = (Y_1, Y_2), \qquad Y_1 \sim f(D), \quad Y_2 \sim g(Y_1, D),$$

satisfies $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.

Combining Theorem 6.2 with Theorem A.1, we obtain the composed RDP profile by summing the per-step costs and then convert it to an (ε, δ)-DP guarantee using the standard RDP-to-DP conversion [18].

We also recall the standard PLD composition rule, which is used to compose the per-step PLDs.

Theorem A.2 (Composition of privacy loss distributions; Theorem 1 in [21]). *For two independent mechanisms with privacy loss distributions ω and ω', the privacy loss distribution of their joint release is given by convolution:*

$$\omega_c = \omega * \omega'.$$

Combining with Theorem 6.3 and Corollary 6.4, we know that the PLD of the full mechanism can be upper bounded by convolving the practical surrogate PLDs.

A.2 Algorithm for Setting Combination Coefficients for LC

We summarize the accounting procedure for LC that output a set of coefficients $\lambda \in \Delta^{N-1}$ in Algorithm 2 below.

Algorithm 2 Setting Combination Coefficients for Linear Combination (LC)

Input: target privacy (ε', δ'), orders grid $\mathcal{A}$ for RDP, hyperparameter sets $\{\mathcal{S}_i\}_{i=1}^N$ of $\{\mathcal{M}_i\}_{i=1}^N$, total step T

Output: set of combination coefficients $\lambda \in \Delta^{N-1}$ satisfying the target privacy.

Case 1: RDP accounting

function DP_EPS((λ, δ))
  for $\alpha \in \mathcal{A}$ **do**
    for $t = 1$ to T **do**
      Compute the per-step RDP parameter $\varepsilon_\alpha^{(t)}(\lambda)$ via (8).
    end for
    $\varepsilon_\alpha^{\mathrm{LC}}(\lambda) \leftarrow \sum_{t=1}^T \varepsilon_\alpha^{(t)}(\lambda)$
    $\varepsilon_\alpha(\lambda; \delta) \leftarrow \varepsilon_\alpha^{\mathrm{LC}}(\lambda) + \frac{\log(1/\delta)}{\alpha - 1}$
  end for
  return: $\min_{\alpha \in \mathcal{A}} \varepsilon_\alpha(\lambda; \delta')$
end function
return $\{\lambda \mid \text{DP_Eps}(\lambda, \delta') \le \varepsilon'\}$

Case 2: PLD accounting

function DP_DELTA((λ, ε))
  for $t = 1$ to T **do**
    Compute the per-step privacy-loss random variable $\tilde{L}^{(t)}$ via Eq. (9)
    Discretize $\tilde{L}^{(t)}$ on grid to obtain its PLD $\omega^{(t)}$
  end for
  Compose the T steps via convolution $\omega \leftarrow \omega^{(1)} * \omega^{(2)} * \cdots * \omega^{(T)}$
  Compute the induced privacy parameter $\delta^{\mathrm{LC}}(\varepsilon)$ from the composed PLD ω.
  return $\delta^{\mathrm{LC}}(\varepsilon)$
end function
return $\{\lambda \mid \text{DP_Delta}(\lambda, \varepsilon') \le \delta'\}$

B Compare with Other Composition Methods

Extensive prior literature has studied DP composition of jointly releasing multiple models [e.g., 7, 12]. Among them, Dwork et al. [7] is one of the seminal works, and Kairouz et al. [12] further improves the bounds. However, these results are stated purely using the privacy information in the form of $(\varepsilon_i, \delta_i)$

parameters of each model. In contrast, assuming access to RDP and PLD parameters typically yields tighter accounting. Hence, the joint-release bounds in Theorem 6.1 (the right-hand sides of Eq (6) and Eq. (7)) using RDP/PLD parameters are in general tighter than (ε, δ)-only composition bounds. Besides, beyond simply assuming access to RDP and PLD parameter, our analysis makes explicit how RS and LC interact with the underlying privacy accounting. RS exploits mixture structure through log-sum-exp and convexity, while LC admits tight bounds when additional algorithmic structure is available. This highlights why our bounds can improve over general joint-release accounting. Compare to joint-release bounds in Theorem 6.1, our LC RDP/PLD accounting gives a tighter result than the joint-release RDP/PLD bound; and RS can also be tighter. Overall, when richer privacy profiles (e.g., RDP parameters across orders) are available, our accounting procedures provide stronger guarantees. Formally, we present Proposition B.1 below, which establishes our argument for the case of N independent and identical Gaussian mechanisms. See complete proofs in Appendix C.8.

Proposition B.1. *Consider releasing N models on the same dataset, where each release follows the same Gaussian mechanism (e.g., mean estimation with sensitivity Δ and noise $Z \sim \mathcal{N}(0, \sigma^2)$). Fix $\delta \in (0, \frac{1}{2})$. By RDP accounting, each model satisfies (ε, δ)-DP. Composing the N releases using the (ε, δ) advanced composition bound [7,* Theorem III.3*] with $\delta' = N\delta + \delta_0$ yields a composed privacy parameter $\varepsilon_{\mathrm{com}}$. Using the joint RDP bound (RHS of Eq. (6)) in Theorem 6.1 yields a guarantee $(\varepsilon_{\mathrm{RDP}}, \delta')$. We have that $\varepsilon_{\mathrm{RDP}} \leq \varepsilon_{\mathrm{com}}$.*

C Complete Proofs

C.1 Proof of Proposition 3.1

Proof. Consider following counterexample: $\theta_1(D) \equiv C$ with probability 1 and then $h(\theta_1, \theta_2)$ can be reduce to a function $\tilde{h}(\theta_2)$ that only depends on θ_2. Since h is nontrival, there at least are two points $\{a, b\}$ s.t $\tilde{h}(a) \overset{d}{\neq} \tilde{h}(b)$. Therefore there is at least one $S \subseteq \mathbb{R}^P$, such that

$$\mathbb{P}(\tilde{h}(a) \in S) > \mathbb{P}(\tilde{h}(b) \in S)$$

Consider $\mathbb{P}(\theta_2(D) = a) = p,\ \mathbb{P}(\theta_2(D) = b) = 1 - p$, and $\mathbb{P}(\theta_2(D') = a) = p',\ \mathbb{P}(\theta_2(D) = b) = 1 - p'$ such that $p = \frac{e^{\varepsilon_2}+\delta}{e^{\varepsilon_2}+1}, p' = \frac{1-\delta}{e^{\varepsilon_2}+1}$. Thus $p = e^{\varepsilon_2}p' + \delta$ and $1 - p' = e^{\varepsilon_2}(1 - p) + \delta$ then

$$\begin{aligned}
\mathbb{P}(\tilde{h}(\theta_2(D)) \in S) &= p\mathbb{P}(\tilde{h}(a) \in S) + (1-p)\mathbb{P}(\tilde{h}(b) \in S) \\
&= e^{\varepsilon_2}p'\mathbb{P}(\tilde{h}(a) \in S) + \delta\mathbb{P}(\tilde{h}(a) \in S) + e^{\varepsilon'}(1-p')\mathbb{P}(\tilde{h}(b) \in S) \\
&\quad - (e^{\varepsilon'} - 1 + (e^{\varepsilon_2} - e^{\varepsilon'})p' + \delta)\mathbb{P}(\tilde{h}(b) \in S) \\
&> e^{\varepsilon'}\mathbb{P}(\tilde{h}(\theta_2(D')) \in S) + ((e^{\varepsilon_2} - e^{\varepsilon'})p' + \delta)\mathbb{P}(\tilde{h}(a) \in S) \\
&\quad - (e^{\varepsilon'} - 1 + (e^{\varepsilon_2} - e^{\varepsilon'})p' + \delta)\mathbb{P}(\tilde{h}(b) \in S) \\
&= e^{\varepsilon'}\mathbb{P}(\tilde{h}(\theta_2(D')) \in S) + \frac{e^{\varepsilon_2} - e^{\varepsilon'} + \delta e^{\varepsilon'} + \delta}{e^{\varepsilon_2}+1}(\mathbb{P}(\tilde{h}(a) \in S) - \mathbb{P}(\tilde{h}(b) \in S)) \\
&\quad - (e^{\varepsilon'} - 1)\mathbb{P}(\tilde{h}(b) \in S)
\end{aligned}$$

The last equality is gained by bringing $p' = e^{-\varepsilon_2}(1 - \delta)$ into the equation.

If $D_{TV}(\tilde{h}(a), \tilde{h}(b)) > \frac{(\delta' + e^{\varepsilon'} - 1)(e^{\varepsilon_2}+1)}{e^{\varepsilon_2+\varepsilon'} + \delta e^{\varepsilon'} + \delta - 1}$, we know

$$\begin{aligned}
&\sup_S \left(\frac{e^{\varepsilon_2} - e^{\varepsilon'} + \delta e^{\varepsilon'} + \delta}{e^{\varepsilon_2}+1}(\mathbb{P}(\tilde{h}(a) \in S) - \mathbb{P}(\tilde{h}(b) \in S)) - (e^{\varepsilon'} - 1)\mathbb{P}(\tilde{h}(b) \in S) \right) \\
&\geq \frac{e^{\varepsilon_2} - e^{\varepsilon'} + \delta e^{\varepsilon'} + \delta}{e^{\varepsilon_2}+1} D_{TV}(\tilde{h}(a), \tilde{h}(b)) - (e^{\varepsilon'} - 1)(1 - D_{TV}(\tilde{h}(a), \tilde{h}(b))) \\
&= \frac{e^{\varepsilon_2+\varepsilon'} + \delta e^{\varepsilon'} + \delta - 1}{e^{\varepsilon_2}+1} D_{TV}(\tilde{h}(a), \tilde{h}(b)) - e^{\varepsilon'} + 1 \\
&> \delta'
\end{aligned}$$

and thus there exists S such that

$$\mathbb{P}(\tilde{h}(\theta_2(D)) \in S) > e^{\varepsilon'}\mathbb{P}(\tilde{h}(\theta_2(D')) \in S) + \delta'.$$

We finish the proof. □

C.2 Proof of Lemma C.1 and Theorem 4.1

Lemma C.1. *For any $\pi, \lambda \in [0,1]$, we have RDP parameter $(\alpha, \varepsilon_\alpha^{RS}(\pi))$ for RS satisfying*

$$\varepsilon_\alpha^{RS} = \mathcal{D}_\alpha\big(P_\pi \| Q_\pi\big) > \frac{\alpha\Delta^2}{2\sigma_{RS}^2(\pi)} + c_\pi\Delta^2 + o(\Delta^2),$$

when $\Delta \to 0$. Here $c_\pi \geq 0$ is a constant depending on π, and when $\pi \in (0,1)$, $c_\pi > 0$. And RDP parameter $(\alpha, \varepsilon_\alpha^{LC}(\lambda))$ for LC satisfying $\varepsilon_\alpha^{LC} = \frac{\alpha\Delta^2}{2\sigma_{LC}^2(\lambda)}$.

Proof of Lemma C.1. Since under linear combination with parameter λ, the output $\theta(D) \sim \mathcal{N}(\hat{\mu}, \lambda^2\sigma_1^2 + (1-\lambda)^2\sigma_2^2)$ and $\sigma_{LC}^2(\lambda) = \lambda^2\sigma_1^2 + (1-\lambda)^2\sigma_2^2$. Thus

$$\varepsilon_\alpha^{\mathrm{LC}} = \frac{\alpha\Delta^2}{2\sigma_{LC}^2(\lambda)}.$$

Next we prove

$$\varepsilon_\alpha^{\mathrm{RS}} = \mathcal{D}_\alpha\big(P_\pi \| Q_\pi\big) \geq \frac{\alpha\Delta^2}{2\sigma_{RS}^2(\pi)} + o(\Delta^2).$$

Write p for the density of P_π and note that Q_π's density $q(x) = p(x-\Delta)$. Define

$$J_\chi(P) \;\coloneqq\; \int_{\mathbb{R}} \frac{(p'(x))^2}{p(x)}\, dx.$$

Step 1. We show that

$$\mathcal{D}_\alpha(P_\pi \| Q_\pi) \;\geq\; \frac{\alpha}{2}\, J_\chi(P_\pi)\, \Delta^2 + o(\Delta^2). \tag{10}$$

Consider $\Psi_\alpha(\Delta) \coloneqq \log\!\int p(x)^\alpha\, p(x-\Delta)^{\,1-\alpha}\, dx$, so that $\mathcal{D}_\alpha(P_\pi \| Q_\pi) = \Psi_\alpha(\Delta)/(\alpha-1)$. Differentiate over Δ and note that $\int p'dx = 0$, we know $\Psi'_\alpha(0) = 0$ and $\Psi''_\alpha(0) = \alpha(\alpha-1)\, J_\chi(P)$, which yields (10).

Step 2. Let random variable $X \sim P_\pi$, we next show

$$J_\chi(P_\pi) \;\geq\; \frac{1}{\mathrm{Var}(X)}. \tag{11}$$

Let $s(x) = p'(x)/p(x)$ be the score. For any smooth h, integration by parts gives $\mathbb{E}[h'(X)] = -\mathbb{E}[h(X)s(X)]$. By Cauchy-Schwarz, $(\mathbb{E}[h'(X)])^2 \leq \mathbb{E}[h(X)^2]\ \mathbb{E}[s(X)^2] = \mathbb{E}[h(X)^2]\ J_\chi(P)$. Taking the supremum over h shows $J_\chi(P) \geq \sup_h \big(\mathbb{E}[h'(X)]^2/\mathbb{E}[h(X)^2]\big)$. Choosing $h(x) = x - \mathbb{E}[X]$ yields $\mathbb{E}[h'(X)] = 1$ and $\mathbb{E}[h(X)^2] = \mathrm{Var}(X)$, proving (11). Note that when $\pi \in (0,1)$, P_π is mixed gaussian distribution, thus the bound could not be reached and thus $c_\pi > 0$.

Combining step 1 and step 2 we finish the proof. □

Proof of Theorem 4.1. Fix $\alpha > 1$ and $\pi \in [0,1]$, and let λ satisfy $\sigma_{\mathrm{LC}}^2(\lambda) = \sigma_{\mathrm{RS}}^2(\pi)$. By Lemma C.1,

$$\varepsilon_\alpha^{\mathrm{LC}}(\lambda) = \frac{\alpha\Delta^2}{2\sigma_{\mathrm{LC}}^2(\lambda)} = \frac{\alpha\Delta^2}{2\sigma_{\mathrm{RS}}^2(\pi)}.$$

Moreover,

$$\varepsilon_\alpha^{\mathrm{RS}}(\pi) > \frac{\alpha\Delta^2}{2\sigma_{\mathrm{RS}}^2(\pi)} + c_\pi\Delta^2 + o(\Delta^2).$$

If $\pi \in (0,1)$, then $c_\pi > 0$, so there exists $C_{\pi,\alpha} > 0$ such that, for all $0 < \Delta < C_{\pi,\alpha}$,

$$\varepsilon_\alpha^{\mathrm{RS}}(\pi) > \varepsilon_\alpha^{\mathrm{LC}}(\lambda).$$

For $\pi \in \{0,1\}$, RS reduces to a single Gaussian mechanism, and the two RDP costs are equal whenever $\sigma_{\mathrm{LC}}^2(\lambda) = \sigma_{\mathrm{RS}}^2(\pi)$. Thus LC achieves privacy no worse than RS.

Finally, choosing such a λ gives $\sigma_{\mathrm{LC}}^2(\lambda) = \sigma_{\mathrm{RS}}^2(\pi)$, so LC has the same MSE as RS and no worse order-α RDP privacy. Hence, under the same target RDP budget, LC achieves no larger MSE. □

C.3 Proof of Theorem 5.1

Lemma C.2. *For any $\alpha > 1$, $\pi \in \Delta^{N-1}$, and $a_i, b_i \geq 0$ ($i = 1, 2, \ldots, N$),*

$$\big(\sum_{i=1}^{N} \pi_i a_i\big)^{\alpha} \big(\sum_{i=1}^{N} \pi_i b_i\big)^{1-\alpha} \;\leq\; \sum_{i=1}^{N} \pi_i a_i^{\alpha} b_i^{1-\alpha}.$$

Proof of C.2. Let the conjugate exponents be $(r, s) = (\alpha, \alpha/(\alpha-1))$. Let $u_i = a_i\, b_i^{(1-\alpha)/\alpha}$, $v_i = b_i^{(\alpha-1)/\alpha}$. By Hölder's inequality, we have

$$\sum_i \pi_i a_i = \sum_i \pi_i u_i v_i \leq \Big(\sum_i \pi_i u_i^r\Big)^{1/r} \Big(\sum_i \pi_i v_i^s\Big)^{1/s} = \Big(\sum_i \pi_i a_i^{\alpha} b_i^{1-\alpha}\Big)^{1/\alpha} \Big(\sum_i \pi_i b_i\Big)^{(\alpha-1)/\alpha}.$$

Raising both sides to the power α and we finish the proof. □ □

Proof of $\varepsilon_{\alpha}^{RS}(\pi)$ upper bound. Let $P_i = \mathsf{Law}(\mathcal{M}_i(D))$ and $Q_i = \mathsf{Law}(\mathcal{M}_i(D'))$ be the output distribution of the two DP mechanisms on neighboring datasets $D \sim D'$. Let

$$P_{\mathrm{mix}} = \sum_{i=1}^{N} \pi_i P_i, \qquad Q_{\mathrm{mix}} = \sum_{i=1}^{N} \pi_i Q_i,$$

be distributions with densities $p_{\mathrm{mix}} = \sum_{i=1}^{N} \pi_i p_i$ and $q_{\mathrm{mix}} = \sum_{i=1}^{N} \pi_i q_i$. By the definition,

$$e^{(\alpha-1)\mathcal{D}_\alpha(P_{\mathrm{mix}}\|Q_{\mathrm{mix}})} = \int p_{\mathrm{mix}}^{\alpha}\, q_{\mathrm{mix}}^{1-\alpha}\, dx = \int \Big(\sum_{i=1}^{N} \pi_i p_i\Big)^{\alpha} \Big(\sum_{i=1}^{N} \pi_i q_i\Big)^{1-\alpha} dx.$$

Applying Lemma C.2, we know

$$p_{\mathrm{mix}}^{\alpha}(x)\, q_{\mathrm{mix}}^{1-\alpha}(x) \leq \sum_{i=1}^{N} \pi_i p_i^{\alpha}(x)\, q_i^{1-\alpha}(x).$$

Thus,

$$e^{(\alpha-1)\mathcal{D}_\alpha(P_{\mathrm{mix}}\|Q_{\mathrm{mix}})} \leq \sum_{i=1}^{N} \pi_i \int p_i^{\alpha} q_i^{1-\alpha} = \sum_{i=1}^{N} \pi_i e^{(\alpha-1)\mathcal{D}_\alpha(P_i\|Q_i)}.$$

Thus,

$$\mathcal{D}_\alpha(P_{\mathrm{mix}}\|Q_{\mathrm{mix}}) \leq \frac{1}{\alpha-1} \log\Big(\sum_{i=1}^{N} \pi_i\, e^{(\alpha-1)\mathcal{D}_\alpha(P_i\|Q_i)}\Big).$$

Since $\varepsilon_{\alpha,i} = \sup_{D\sim D'} \mathcal{D}_\alpha(P_i\|Q_i)$, we obtain

$$\varepsilon_{\alpha}^{\mathsf{RS}}(\pi) \leq \frac{1}{\alpha-1} \log\Big(\sum_{i=1}^{N} \pi_i\, e^{(\alpha-1)\varepsilon_{\alpha,i}}\Big), \quad \forall \alpha > 1.$$

We finish the proof. □

C.4 Proof of Theorem 5.2

Proof of Theorem 5.2. For any $\varepsilon \geq 0$, recall the hockey-stick divergence is defined as

$$\mathcal{H}_\varepsilon(P, Q) = \sup_S \Big\{P(S) - e^{\varepsilon} Q(S)\Big\}, \tag{12}$$

Thus for any measurable S,

$$P_{RS}(S) - e^{\varepsilon} Q_{RS}(S) = \sum_{i=1}^{N} \pi_i \Big(P_i(S) - e^{\varepsilon} Q_i(S)\Big).$$

Taking supremum over S, we obtain

$$\begin{aligned}\mathcal{H}_\varepsilon(P_{RS}, Q_{RS}) &= \sup_A \sum_{i=1}^N \pi_i \Big(P_i(A) - e^\varepsilon Q_i(A) \Big) \\ &\le \sum_{i=1}^N \pi_i \sup_A \Big(P_i(A) - e^\varepsilon Q_i(A) \Big) = \sum_{i=1}^N \pi_i \mathcal{H}_\varepsilon(P_i, Q_i).\end{aligned}$$

This proves the first inequality. The second inequality follows by the same argument after swapping the roles of P and Q:

$$\mathcal{H}_\varepsilon(Q_{RS}, P_{RS}) \le \sum_{i=1}^N \pi_i \mathcal{H}_\varepsilon(Q_i, P_i).$$

Using the two bounds above for each neighboring pair $D \sim D'$, we have

$$\delta^{\mathrm{RS}}(\varepsilon) \le \sup_{D \sim D'} \max \Big\{ \sum_{i=1}^N \pi_i \, \mathcal{H}_\varepsilon(P_i, Q_i), \; \sum_{i=1}^N \pi_i \, \mathcal{H}_\varepsilon(Q_i, P_i) \Big\},$$

We finish the proof. □

C.5 Proof of Theorem 6.1

Proof. Fix any neighboring datasets $D \sim D'$, let $P^{\mathrm{joint}}, Q^{\mathrm{joint}}$ be the output distributions of the joint release $\mathcal{M}_{\mathrm{joint}} := (\mathcal{M}_1, \ldots, \mathcal{M}_N)$ on D and D', respectively. Similarly, we define $P^{\mathrm{LC}}, Q^{\mathrm{LC}}$ be the output distributions of LC on D and D'. Obviously, the linear-combination mechanism can be understood as a post-processing of the joint release.

(1) RDP upper bound. Since the linear-combination mechanism can be understood as a post-processing of the joint release, we have

$$D_\alpha(P^{\mathrm{LC}} \,\|\, Q^{\mathrm{LC}}) \le D_\alpha(P^{\mathrm{joint}} \,\|\, Q^{\mathrm{joint}}). \tag{13}$$

Under the definition of Rényi divergence, we have

$$D_\alpha(P^{\mathrm{joint}} \,\|\, Q^{\mathrm{joint}}) = \sum_{i=1}^N D_\alpha(P_i \,\|\, Q_i) \le \sum_{i=1}^N \varepsilon_{\alpha,i}. \tag{14}$$

Combining (13)-(14) and taking the worst case over $D \sim D'$ yields $\varepsilon_\alpha^{\mathrm{LC}}(\lambda) \le \sum_{i=1}^N \varepsilon_{\alpha,i}$.

(2) PLD upper bound. Since the linear-combination mechanism can be understood as a post-processing of the joint release, we have

$$\mathcal{H}_\varepsilon(P^{\mathrm{LC}} \,\|\, Q^{\mathrm{LC}}) \le \mathcal{H}_\varepsilon(P^{\mathrm{joint}} \,\|\, Q^{\mathrm{joint}}). \tag{15}$$

Similarly, we have

$$\mathcal{H}_\varepsilon(Q^{\mathrm{LC}}, P^{\mathrm{LC}}) \le \mathcal{H}_\varepsilon(Q^{\mathrm{joint}}, P^{\mathrm{joint}}).$$

Therefore,

$$\begin{aligned}\delta^{\mathrm{LC}}(\varepsilon) &= \max\{\mathcal{H}_\varepsilon(P^{\mathrm{LC}}, Q^{\mathrm{LC}}), \mathcal{H}_\varepsilon(Q^{\mathrm{LC}}, P^{\mathrm{LC}})\} \\ &\le \max\{\mathcal{H}_\varepsilon(P^{\mathrm{joint}}, Q^{\mathrm{joint}}), \mathcal{H}_\varepsilon(Q^{\mathrm{joint}}, P^{\mathrm{joint}})\} =: \delta_{\mathrm{joint}}(\varepsilon).\end{aligned}$$

Moreover, from PLD accounting, we know the PLD of the joint release is the convolution $\omega_{\mathrm{joint}} = \omega_1 \star \cdots \star \omega_N$, and $\delta_{\mathrm{joint}}(\varepsilon)$ is computed from ω_{joint}.

(3) Tightness. We show the bounds in (6)-(7) are worst-case tight by an explicit construction. Let $p_1, \ldots, p_N$ be positive integers and set $p = \sum_{i=1}^N p_i$. For each $i \in [N]$, let $\mathcal{M}_i'$ be any mechanism whose output lies in $\mathbb{R}^{p_i}$, and $\mathcal{M}_i$ output its p dimensional version $\theta_i : \mathcal{D} \to \mathbb{R}^p$, where the $\sum_{j=1}^{i-1} p_j + 1-$th element to $\sum_{j=1}^{i} p_j-$th element is $M_i'(D)$ and other elements are 0.

Now consider the linear-combination output

$$\theta(D) \ := \ \sum_{i=1}^N \lambda_i \theta_i(D) \in \mathbb{R}^p.$$

Intuitively, linear-combination does not lose any information from the joint release. As a result, for every neighboring pair $D \sim D'$ and every $\alpha > 1$, Rényi divergence is preserved. We have

$$D_\alpha\big(P^{\mathrm{LC}} \,\|\, Q^{\mathrm{LC}}\big) = D_\alpha\big(P_{\mathrm{joint}} \,\|\, Q_{\mathrm{joint}}\big),$$

and likewise the hockey-stick divergence (hence the PLD profile) is also preserved:

$$\delta^{\mathrm{LC}}(\varepsilon) = \delta_{\mathrm{joint}}(\varepsilon), \qquad \forall\, \varepsilon \geq 0.$$

Therefore the upper bounds (6)-(7) cannot be improved in general. This establishes worst-case tightness. □

C.6 Proof of Theorem 6.2

Proof of Theorem 6.2. We first establish a uniform conditional RDP bound for one step. Let D, D' be neighboring datasets under replace-one adjacency. Let H and H' denote arbitrary possible value of $(\theta_1^{(t-1)}, \cdots, \theta_N^{(t-1)})$ before step t under D and D'.

$$\theta^{(t)} - \theta^{(t-1)} = \sum_{i=1}^N \lambda_i \eta_i^{(t)} \widetilde{g}_i^{(t)}(D; H) + \overline{e}_\lambda^{(t)},$$

where $\widetilde{g}_i^{(t)}(D; H)$ is clipped gradient for model i and

$$\overline{e}_\lambda^{(t)} = \sum_{i=1}^N \lambda_i \eta_i^{(t)} e_i^{(t)} \sim \mathcal{N}(0, (s_\lambda^{(t)})^2 I), \qquad s_\lambda^2 = \sum_{i=1}^N (\lambda_i \eta_i^{(t)} \sigma_i^{(t)} C_i^{(t)})^2.$$

Thus, conditional on H, the distribution of $\theta^{(t)} - \theta^{(t-1)}$ under D is Gaussian with covariance $(s_\lambda^{(t)})^2 I$. Similarly, conditional on H', the distribution under D' is Gaussian with the same covariance.

By clipping,

$$\left\| \sum_{i=1}^N \lambda_i \eta_i^{(t)} \left(\widetilde{g}_i^{(t)}(D; H) - \widetilde{g}_i^{(t)}(D'; H') \right) \right\|_2 \leq 2 \sum_{i=1}^N \lambda_i \eta_i^{(t)} C_i^{(t)} = \Delta_\lambda^{(t)}.$$

Hence, for any pair of H, H', the RDP between the two corresponding Gaussian conditional distributions is bounded by

$$D_\alpha \left(\mathcal{N}(\sum_{i=1}^N \lambda_i \eta_i^{(t)} \widetilde{g}_i^{(t)}(D), (s_\lambda^{(t)})^2 I) \middle\| \mathcal{N}(\sum_{i=1}^N \lambda_i \eta_i^{(t)} \widetilde{g}_i^{(t)}(D'), (s_\lambda^{(t)})^2 I) \right) \leq \frac{\alpha (\Delta_\lambda^{(t)})^2}{2 (s_\lambda^{(t)})^2}.$$

Recall

$$\mathcal{F}_{t-1} = \sigma(\theta^{(t-1)}).$$

The distribution of $\theta^{(t)} - \theta^{(t-1)}$ conditional on $\mathcal{F}_{t-1}$ under D can be written as a mixture over H:

$$\mathbb{P}_{\theta^{(t)} - \theta^{(t-1)}}(\cdot \mid D, \mathcal{F}_{t-1}) = \int \mathbb{P}_{\theta^{(t)} - \theta^{(t-1)}}(\cdot \mid D, H) \, d\kappa(H \mid D, F_{t-1}),$$

and similarly,

$$\mathbb{P}_{\theta^{(t)} - \theta^{(t-1)}}(\cdot \mid D', \mathcal{F}_{t-1}) = \int \mathbb{P}_{\theta^{(t)} - \theta^{(t-1)}}(\cdot \mid D, H') \, d\kappa(H' \mid D', F_{t-1}),$$

where $\kappa(\cdot \mid D, \mathcal{F}_{t-1})$ denotes the regular conditional distribution of H and H' Similar to Theorem 5.1, using joint convexity RDP, let P, Q be the distribution of $\theta^{(t)} - \theta^{(t-1)}$ conditional on $\mathcal{F}_{t-1}$ under D and D' respectively, and we obtain

$$D_\alpha(P\|Q) \leq \frac{\alpha(\Delta_\lambda^{(t)})^2}{2(s_\lambda^{(t)})^2}.$$

□

C.7 Proof of Theorem 6.3 and Corollary 6.4

Proof of Theorem 6.3. Let D, D' be neighboring datasets under replace-one adjacency. Let H and H' denote arbitrary possible value of $(\theta_1^{(t-1)}, \cdots, \theta_N^{(t-1)})$ before step t under D and D', respectively. Conditional on H, the combined update has the form

$$\theta^{(t)} - \theta^{(t-1)} = \sum_{i=1}^N \lambda_i \eta_i^{(t)} \widetilde{g}_i^{(t)}(D; H) + \overline{e}_\lambda^{(t)},$$

where $\widetilde{g}_i(D; H)$ is clipped gradient for model i and

$$\overline{e}_\lambda^{(t)} = \sum_{i=1}^N \lambda_i \eta_i^{(t)} e_i^{(t)} \sim \mathcal{N}(0, (s_\lambda^{(t)})^2 I), \qquad s_\lambda^2 = \sum_{i=1}^N (\lambda_i \eta_i^{(t)} \sigma_i^{(t)} C_i^{(t)})^2.$$

Thus, conditional on H, the distribution of $\theta^{(t)} - \theta^{(t-1)}$ under D is Gaussian with covariance $(s_\lambda^{(t)})^2 I$. Similarly, conditional on H', the distribution under D' is Gaussian with the same covariance. By clipping,

$$\left\| \sum_{i=1}^N \lambda_i \eta_i^{(t)} \left(\widetilde{g}_i^{(t)}(D; H) - \widetilde{g}_i^{(t)}(D'; H') \right) \right\|_2 \leq 2 \sum_{i=1}^N \lambda_i \eta_i^{(t)} C_i^{(t)} = \Delta_\lambda^{(t)}.$$

We first bound the hockey-stick divergence conditional on H and H'. Conditioned on H, H' and the neighboring datasets D, D', the two conditional distributions differ only through their Gaussian means and share the same covariance $(s_\lambda^{(t)})^2 I$. Let

$$v = v(H, H', D, D')$$

denote the corresponding mean-difference vector. By the preceding sensitivity bound,

$$\|v\|_2 \leq \Delta_\lambda^{(t)}.$$

Let $F_{D,H}$ and $F_{D',H'}$ denote the conditional distributions of $\theta^{(t)} - \theta^{(t-1)}$ under (D, H) and (D', H'), respectively. By translation invariance, the privacy loss random variable for the pair $(F_{D,H}, F_{D',H'})$ can be written as

$$L_v^{(t)} = \log \frac{\phi(Y; v, (s_\lambda^{(t)})^2 I)}{\phi(Y; 0, (s_\lambda^{(t)})^2 I)} = \frac{\langle Y, v \rangle}{(s_\lambda^{(t)})^2} - \frac{\|v\|_2^2}{2(s_\lambda^{(t)})^2}, \qquad Y \sim \mathcal{N}(0, (s_\lambda^{(t)})^2 I).$$

Equivalently, letting $Z \sim \mathcal{N}(0, (s_\lambda^{(t)})^2)$ and $r = \|v\|_2$, we have

$$L_v^{(t)} \overset{d}{=} \frac{rZ}{(s_\lambda^{(t)})^2} - \frac{r^2}{2(s_\lambda^{(t)})^2} = \log \frac{\phi(Z; r, (s_\lambda^{(t)})^2)}{\phi(Z; 0, (s_\lambda^{(t)})^2)},$$

where ϕ denotes the one-dimensional Gaussian density.

Therefore, by the definition of the hockey-stick divergence,

$$\mathcal{H}_\varepsilon(F_{D,H}, F_{D',H'}) = \mathbb{E}\left[(e^{L_v^{(t)}} - e^\varepsilon)_+ \right],$$

and, for the reverse direction,

$$\mathcal{H}_\varepsilon(F_{D',H'}, F_{D,H}) = \mathbb{E}\Big[(1 - e^{\varepsilon + L_v^{(t)}})_+\Big].$$

Let $s = s_\lambda^{(t)}$ and, for $r \geq 0$, define

$$L_r = \frac{rZ}{s^2} - \frac{r^2}{2s^2}, \qquad Z \sim \mathcal{N}(0, s^2).$$

We first show that $\mathbb{E}\big[(e^{L_r} - e^\varepsilon)_+\big]$ is nondecreasing in r. For $r > 0$,

$$e^{L_r} > e^\varepsilon \iff Z > a_r, \qquad a_r := \frac{\varepsilon s^2}{r} + \frac{r}{2}.$$

Therefore,

$$\mathbb{E}\big[(e^{L_r} - e^\varepsilon)_+\big] = \int_{a_r}^\infty \left(e^{\frac{rz}{s^2} - \frac{r^2}{2s^2}} - e^\varepsilon\right) \phi(z; 0, s^2)\, dz.$$

Using

$$e^{\frac{rz}{s^2} - \frac{r^2}{2s^2}} \phi(z; 0, s^2) = \phi(z; r, s^2),$$

we can rewrite this as

$$\mathbb{E}\big[(e^{L_r} - e^\varepsilon)_+\big] = \int_{a_r}^\infty \big(\phi(z; r, s^2) - e^\varepsilon \phi(z; 0, s^2)\big)\, dz.$$

Differentiating with respect to r, the boundary term vanishes because the integrand is zero at $z = a_r$. Thus,

$$\frac{d}{dr}\mathbb{E}\big[(e^{L_r} - e^\varepsilon)_+\big] = \int_{a_r}^\infty \frac{\partial}{\partial r}\phi(z; r, s^2)\, dz.$$

Since

$$\frac{\partial}{\partial r}\phi(z; r, s^2) = -\frac{\partial}{\partial z}\phi(z; r, s^2),$$

we obtain

$$\frac{d}{dr}\mathbb{E}\big[(e^{L_r} - e^\varepsilon)_+\big] = \phi(a_r; r, s^2) \geq 0.$$

Hence $\mathbb{E}[(e^{L_r} - e^\varepsilon)_+]$ is nondecreasing in r.

Similarly, we show that $\mathbb{E}\big[(1 - e^{\varepsilon + L_r})_+\big]$ is nondecreasing in r. Since

$$1 - e^{\varepsilon + L_r} > 0 \iff L_r < -\varepsilon \iff \frac{rZ}{s^2} - \frac{r^2}{2s^2} < -\varepsilon,$$

for $r > 0$ this is equivalent to

$$Z < b_r, \qquad b_r := \frac{r}{2} - \frac{\varepsilon s^2}{r}.$$

Therefore,

$$\mathbb{E}\big[(1 - e^{\varepsilon + L_r})_+\big] = \int_{-\infty}^{b_r} \left(1 - e^\varepsilon e^{\frac{rz}{s^2} - \frac{r^2}{2s^2}}\right) \phi(z; 0, s^2)\, dz.$$

Using

$$e^{\frac{rz}{s^2} - \frac{r^2}{2s^2}} \phi(z; 0, s^2) = \phi(z; r, s^2),$$

we can rewrite this as

$$\mathbb{E}\big[(1 - e^{\varepsilon + L_r})_+\big] = \int_{-\infty}^{b_r} \big(\phi(z; 0, s^2) - e^\varepsilon \phi(z; r, s^2)\big)\, dz.$$

Differentiating with respect to r, the boundary term vanishes because the integrand is zero at $z = b_r$. Thus,

$$\frac{d}{dr}\mathbb{E}\big[(1 - e^{\varepsilon + L_r})_+\big] = -e^\varepsilon \int_{-\infty}^{b_r} \frac{\partial}{\partial r}\phi(z; r, s^2)\, dz.$$

Since

$$\frac{\partial}{\partial r}\phi(z;r,s^2) = -\frac{\partial}{\partial z}\phi(z;r,s^2),$$

we obtain

$$\frac{d}{dr}\mathbb{E}\big[(1-e^{\varepsilon+L_r})_+\big] = e^{\varepsilon}\int_{-\infty}^{b_r}\frac{\partial}{\partial z}\phi(z;r,s^2)\,dz = e^{\varepsilon}\phi(b_r;r,s^2) \geq 0.$$

Hence,

$$\mathbb{E}\big[(1-e^{\varepsilon+L_r})_+\big]$$

is also nondecreasing in r.

Recall $r \leq \Delta_\lambda^{(t)}$, $\mathbb{E}\big[(1-e^{\varepsilon+L_r})_+\big], \mathbb{E}[(e^{L_r}-e^{\varepsilon})_+]$ are upper bounded by the corresponding quantities with $r = \Delta_\lambda^{(t)}$.

Since $L_r \overset{d}{=} L_v^{(t)}$, we know

$$\mathbb{E}\big[(1-e^{\varepsilon+L_r})_+\big] = \mathbb{E}\Big[(1-e^{\varepsilon+L_v^{(t)}})_+\Big], \qquad \mathbb{E}[(e^{L_r}-e^{\varepsilon})_+] = \mathbb{E}[(e^{L_v^{(t)}}-e^{\varepsilon})_+]$$

.

Hence, defining

$$\widetilde{L}^{(t)} = \log\frac{\phi(Z;\Delta_\lambda^{(t)},(s_\lambda^{(t)})^2)}{\phi(Z;0,(s_\lambda^{(t)})^2)}, \qquad Z \sim \mathcal{N}(0,(s_\lambda^{(t)})^2),$$

we obtain, uniformly over all H, H', D, D',

$$\mathcal{H}_\varepsilon(F_{D,H}, F_{D',H'}) \leq \mathbb{E}\Big[(e^{\widetilde{L}^{(t)}}-e^{\varepsilon})_+\Big],$$

and

$$\mathcal{H}_\varepsilon(F_{D',H'}, F_{D,H}) \leq \mathbb{E}\Big[(1-e^{\varepsilon+\widetilde{L}^{(t)}})_+\Big].$$

Now condition on the $\mathcal{F}_{t-1}$. The conditional distribution under D can be written as a mixture over H and H':

$$P^{(t)} = \int F_{D,H}\,d\kappa(H \mid D, \mathcal{F}_{t-1}),$$

and similarly

$$Q^{(t)} = \int F_{D',H'}\,d\kappa(H' \mid D', \mathcal{F}_{t-1}),$$

where $\kappa(\cdot \mid D, \mathcal{F}_{t-1})$ denotes the regular conditional distribution of the hidden states given the previously accounted transcript under dataset D.

By joint convexity of the hockey-stick divergence,

$$\mathcal{H}_\varepsilon(P^{(t)}, Q^{(t)}) \leq \iint \mathcal{H}_\varepsilon\left(F_{D,H}, F_{D',H'}\right)\,d\kappa(H \mid D, \mathcal{F}_{t-1})\,d\kappa(H' \mid D', \mathcal{F}_{t-1}).$$

Therefore,

$$\mathcal{H}_\varepsilon(P^{(t)}, Q^{(t)}) \leq \mathbb{E}[(e^{\widetilde{L}^{(t)}}-e^{\varepsilon})_+].$$

The same argument with the two distributions swapped gives

$$\mathcal{H}_\varepsilon(Q^{(t)}, P^{(t)}) \leq \mathbb{E}[(1-e^{\varepsilon+\widetilde{L}^{(t)}})_+].$$

This proves the theorem. □

Proof of Corollary 6.4. We consider the following lemma.

Lemma C.3. *Let $L^{(t)}$ be privacy loss random variables step t ($t \in [1 \dots T]$), conditioned on $\mathcal{F}_{t-1}$, and $\tilde{L}^{(t)}$ is surrogate. Define*

$$\delta_{L^{(t)}}(\varepsilon) := \mathbb{E}[(1 - e^{\varepsilon - L^{(t)}})_+], \qquad \delta_{\tilde{L}^{(t)}}(\varepsilon) := \mathbb{E}\big[(1 - e^{\varepsilon - \tilde{L}^{(t)}})_+\big].$$

Suppose that $\forall t$

$$\delta_{\tilde{L}^{(t)}}(\varepsilon) \geq \delta_{L^{(t)}}(\varepsilon), \qquad \forall \varepsilon \in \mathbb{R}.$$

Then

$$\delta_{\tilde{L}^{(1)}+\cdots+\tilde{L}^{(T)}}(\varepsilon) \geq \delta_{L^{(1)}+\cdots+L^{(T)}}(\varepsilon), \qquad \forall \varepsilon \in \mathbb{R}.$$

Proof. For $T = 2$, note

$$\begin{aligned}\delta_{L^{(1)}+L^{(2)}}(\varepsilon) &= \mathbb{E}[e^{L^{(1)}}\mathbb{E}[(e^{L^{(2)}} - e^{\varepsilon - L^{(1)}})^+] \mid F_1] \\ &= \mathbb{E}[e^{L^{(1)}}\delta_{L^{(2)}}(\varepsilon - L^{(1)})] \leq \mathbb{E}[e^{L^{(1)}}\delta_{\tilde{L}^{(2)}}(\varepsilon - L^{(1)})] = \delta_{L^{(1)}+\tilde{L}^{(2)}}(\varepsilon).\end{aligned}$$

By the construction, $\tilde{L}^{(2)}$ only depend on randomness in step 2 and thus is independent to $L^{(1)}$. So similarly, $\delta_{L^{(1)}+\tilde{L}^{(2)}}(\varepsilon) \leq \delta_{\tilde{L}^{(1)}+\tilde{L}^{(2)}}(\varepsilon)$. Thus

$$\delta_{L^{(1)}+L^{(2)}}(\varepsilon) \leq \delta_{\tilde{L}^{(1)}+\tilde{L}^{(2)}}(\varepsilon).$$

This proves the result for $T = 2$.

Assume now that the statement holds for $T = k - 1$. For $T = k$, let

$$O_1 = L^{(1)} + \cdots + L^{(k-1)}, \qquad O_2 = L^{(k)},$$

and define O_1', O_2' analogously. By the induction hypothesis, $\delta_{O_1'}(\varepsilon) \geq \delta_{O_1}(\varepsilon)$ for all ε. Applying the two-term result to (O_1, O_2) and (O_1', O_2') yields the claim for $T = k$. By induction, we finish the proof. □

Corollary 6.4 is a direct application of Lemma C.3. We finish the proof. □

C.8 Proof of Proposition B.1

Proof. WLOG, we assume $\frac{\Delta}{\sigma} = 1$

Step 1 Per-release RDP parameter. For the Gaussian mechanism (mean estimation with ℓ_2-sensitivity Δ and noise $Z \sim \mathcal{N}(0, \sigma^2)$), the per-release Rényi DP parameter at order $\alpha > 1$ is

$$\varepsilon_\alpha \;=\; \frac{\alpha\,\Delta^2}{2\sigma^2}.$$

Step 2 Per-release (ε, δ) from RDP. Since

$$\varepsilon \;=\; \inf_{\alpha > 1}\left\{\varepsilon_\alpha + \frac{\log(1/\delta)}{\alpha - 1}\right\},$$

We know

$$\varepsilon = \frac{\Delta^2}{2\sigma^2} + \frac{\Delta}{\sigma}\sqrt{2\log(1/\delta)}$$

Step 3 composition in (ε, δ). Applying [7, Theorem III.3] to N releases, each satisfying (ε, δ)-DP, yields a composed guarantee $(\varepsilon_{\mathrm{com}}, \delta')$ with

$$\delta' \;=\; N\delta + \delta_0,$$

and

$$\varepsilon_{\mathrm{com}} = \varepsilon\sqrt{2N\log(1/\delta_0)} + N\varepsilon(e^\varepsilon - 1)$$

.

Step 4 Joint RDP accounting. By additivity of RDP under composition, we know

$$\varepsilon_{\mathrm{RDP}} \;=\; \inf_{\alpha > 1}\left\{N\varepsilon_\alpha + \frac{\log(1/\delta')}{\alpha - 1}\right\} = \inf_{\alpha > 1}\left\{\frac{N\alpha\,\Delta^2}{2\sigma^2} + \frac{\log(1/\delta')}{\alpha - 1}\right\}$$

Thus

$$\varepsilon_{\mathrm{RDP}} = \frac{N\Delta^2}{2\sigma^2} + \frac{\Delta}{\sigma}\sqrt{2N\log(1/\delta')}.$$

We only need to prove

$$\frac{N\Delta^2}{2\sigma^2} + \frac{\Delta}{\sigma}\sqrt{2N\log(1/\delta')} \le \varepsilon\sqrt{2N\log(1/\delta_0)} + N\varepsilon(e^\varepsilon - 1).$$

Let $t = \frac{\Delta}{\sigma}, s = \sqrt{2\log(1/\delta)}$. Bringing

$$\varepsilon = \frac{1}{2}t^2 + st$$

into the inequality, and use $e^\varepsilon > 1 + \varepsilon$, since $e^\varepsilon > 1 + \varepsilon$, we only need to prove

$$\frac{N}{2}t^2 + t\sqrt{2N\log(1/\delta')} \le (\frac{1}{2}t^2 + st)\sqrt{2N\log(1/\delta_0)} + N(\frac{1}{2}t^2 + st)^2$$

Let $A \triangleq \sqrt{2N\log(1/\delta_0)}$ and $C \triangleq \sqrt{2N\log(1/\delta')}$. Since $\delta' = N\delta + \delta_0 \ge \delta_0$, we have $C \le A$. Therefore the left-hand side is at most $\frac{N}{2}t^2 + tA$, and it is enough to show

$$\frac{N}{2}t^2 + tA \;\le\; \Big(\frac{1}{2}t^2 + st\Big)A \;+\; N\Big(\frac{1}{2}t^2 + st\Big)^2. \tag{16}$$

Rearranging (16), the gap equals

$$\begin{aligned}&\Big(\frac{1}{2}t^2 + st - t\Big)A \;+\; N\Big[\Big(\frac{1}{2}t^2 + st\Big)^2 - \frac{1}{2}t^2\Big]\\ &= \Big(\frac{1}{2}t^2 + (s-1)t\Big)A \;+\; N\Big(\frac{1}{4}t^4 + st^3 + (s^2 - \frac{1}{2})t^2\Big).\end{aligned}$$

Since $\delta \le 1/2$, we have $s = \sqrt{2\log(1/\delta)} \ge \sqrt{2\log 2} > 1$, hence both $s - 1 \ge 0$ and $s^2 - \frac{1}{2} \ge 0$. Since $t \ge 0$ and $A \ge 0$, every term on the right-hand side is nonnegative, so the gap is ≥ 0. We finish the proof. □

C.9 Proof of Proposition 6.5

Let $D \sim D'$ be neighboring datasets and let $\Theta = (\theta_1(D), \dots, \theta_N(D))$ denote the input models. For $\lambda \in \Lambda$, write $\theta_\lambda(D)$ for the merged model (for RS, $\theta_\lambda(D)$ is itself randomized through the selection index), and let $P_\lambda = \mathrm{Law}(\theta_\lambda(D))$ and $Q_\lambda = \mathrm{Law}(\theta_\lambda(D'))$; by assumption, $\mathcal{H}_{\varepsilon'}(P_\lambda, Q_\lambda) \le \delta'$ and $\mathcal{H}_{\varepsilon'}(Q_\lambda, P_\lambda) \le \delta'$ for every $\lambda \in \Lambda$. The mechanism of Proposition 6.5 computes $s_\lambda = s_\lambda(\Theta, D) \in [0, 1]$, draws λ^* with probability

$$\rho_\lambda = \frac{e^{ts_\lambda}}{\sum_{\lambda' \in \Lambda} e^{ts_{\lambda'}}}, \qquad \lambda \in \Lambda,$$

and releases $\theta_{\lambda^*}(D)$. Let P_{sel} and Q_{sel} denote the output distributions of this mechanism on D and D', and let $\overline{P}$ and $\overline{Q}$ denote the output distributions on D and D' of the auxiliary mechanism that instead draws λ^* uniformly from Λ, independently of Θ and the data, i.e., $\overline{P} = \frac{1}{K}\sum_{\lambda \in \Lambda} P_\lambda$ and $\overline{Q} = \frac{1}{K}\sum_{\lambda \in \Lambda} Q_\lambda$.

Proof of Proposition 6.5. **Step 1: ρ is close to uniform.** Let

$$t_1 = \log\frac{Ke^t}{e^t + K - 1}, \qquad t_2 = \log\frac{(K-1)e^t + 1}{K},$$

so that $0 \le t_1, t_2 \le t$. Since $s_\lambda \in [0, 1]$, we have $e^{ts_\lambda} \in [1, e^t]$ for every λ; hence the numerator of ρ_λ is at most e^t while the denominator is at least $e^t + (K - 1)$, and the numerator is at least 1 while the denominator is at most $1 + (K - 1)e^t$. Therefore, uniformly over Θ, the data, and $\lambda \in \Lambda$,

$$\frac{e^{-t_2}}{K} = \frac{1}{(K-1)e^t + 1} \;\le\; \rho_\lambda \;\le\; \frac{e^t}{e^t + K - 1} = \frac{e^{t_1}}{K}. \tag{17}$$

Step 2: comparison with the uniform mixture. Fix a measurable set S of outputs. Conditioning on Θ and then on λ^*, and writing $\mathbb{E}_D$ for the expectation over the training randomness of Θ on D,

$$P_{\mathrm{sel}}(S) = \mathbb{E}_D\Big[\sum_{\lambda\in\Lambda} \rho_\lambda\, \mathbb{P}\big(\theta_\lambda(D)\in S \,\big|\, \Theta\big)\Big] \le e^{t_1}\, \mathbb{E}_D\Big[\sum_{\lambda\in\Lambda} \frac{1}{K}\, \mathbb{P}\big(\theta_\lambda(D)\in S \,\big|\, \Theta\big)\Big] = e^{t_1}\, \overline{P}(S),$$

where the inequality uses the upper bound in (17) (for LC, $\mathbb{P}(\theta_\lambda(D)\in S \mid \Theta)$ is simply the indicator of $\theta_\lambda(D)\in S$). The same computation on D' with the lower bound in (17) gives $Q_{\mathrm{sel}}(S) \ge e^{-t_2}\,\overline{Q}(S)$, i.e.,

$$\overline{Q}(S) \le e^{t_2}\, Q_{\mathrm{sel}}(S).$$

Step 3: the uniform mixture is (ε', δ')-DP. The auxiliary mechanism is exactly random selection (Eq. (4)) over the K mechanisms $\{\theta_\lambda(\cdot)\}_{\lambda\in\Lambda}$ with data-independent weights $\pi_\lambda = 1/K$. By Theorem 5.2,

$$\overline{P}(S) - e^{\varepsilon'}\,\overline{Q}(S) \le \mathcal{H}_{\varepsilon'}(\overline{P}, \overline{Q}) \le \frac{1}{K}\sum_{\lambda\in\Lambda} \mathcal{H}_{\varepsilon'}(P_\lambda, Q_\lambda) \le \delta'.$$

Step 4: conclusion. Combining Steps 2 and 3,

$$P_{\mathrm{sel}}(S) \le e^{t_1}\,\overline{P}(S) \le e^{t_1}\big(e^{\varepsilon'}\,\overline{Q}(S) + \delta'\big) \le e^{\varepsilon'+t_1+t_2}\, Q_{\mathrm{sel}}(S) + e^{t_1}\delta' \le e^{\varepsilon'+2t}\, Q_{\mathrm{sel}}(S) + e^{t}\delta',$$

where the last step uses $t_1, t_2 \le t$. Since S was arbitrary and the roles of D and D' can be exchanged, the mechanism is $(\varepsilon' + 2t,\ e^t\delta')$-DP. We finish the proof. □

Remark C.4 (Sharper constants). The proof actually gives $(\varepsilon' + t_1 + t_2,\ e^{t_1}\delta')$-DP with

$$t_1 + t_2 = \log \frac{e^t\big((K-1)e^t + 1\big)}{e^t + K - 1} \ \le\ 2t,$$

since $(K-1)e^t + 1 \le e^t(e^t + K - 1)$. For $K = 2$ one has $t_1 + t_2 = t$; as $K \to \infty$, both t_1 and t_2 increase to t, so $2t$ is the worst case over grid sizes.

D Experiment Details

We use 42 for random seed in all experiments. The main experiments were run on a single NVIDIA RTX 4090 GPU, and the total runtime varied from a few hours to one day across different datasets and model architectures.

D.1 MNIST

Dataset and Model. We use MNIST with the standard train/test split (60,000/10,000). Images are normalized. We train a logistic regression classifier:

$$\texttt{Flatten} \to \texttt{Linear}(28\times 28 \to 10),$$

optimized with cross-entropy loss. We report test-set classification accuracy (and loss) after each epoch and use the final test accuracy for Pareto-frontier plots.

DP-SGD Implementation. We use the Opacus package [28] to train with DP-SGD. The optimizer is SGD. In our experiments, we train three base models with the following hyperparameters:

$$\text{max learning rate } 4, \quad C = 2, \quad \sigma = 32, \quad T = 20 \text{ epochs};$$

$$\text{max learning rate } 4, \quad C = 4, \quad \sigma = 32, \quad T = 20 \text{ epochs};$$

$$\text{max learning rate } 4, \quad C = 2, \quad \sigma = 64, \quad T = 20 \text{ epochs}.$$

We use $N = 60{,}000$ training samples and full batch. We use a linear learning-rate schedule with a 10%-epoch warmup followed by linear decay.

D.2 CIFAR-10

Dataset and Model. We use CIFAR-10 with the standard split ($50{,}000/10{,}000$). We apply standard data preprocessing. We use ResNet18 with CIFAR-10 adaptations: the first convolution is replaced by a 3×3 stride-1 convolution and the max-pooling layer is removed. The final fully-connected layer is set to output 10 classes. We report test-set accuracy (and loss) after each epoch; the final test accuracy is used for the accuracy–privacy Pareto-frontier plots.

DP-SGD from a Pretrained Model Initialization. In addition to training from scratch, we also consider a pretraining-based setup on CIFAR-10 to better reflect practical workflows. We randomly split the original training set into two disjoint subsets of equal size, denoted by D_{pre} and D_{priv}. We first train a non-private ResNet18 model on D_{pre} using standard SGD, and then use the resulting checkpoint as initialization for DP-SGD on D_{priv}. All merged candidate models are produced from this common pretrained initialization. The privacy guarantee in this setting is with respect to D_{priv} only, since D_{pre} is disjoint from the private training subset and is not included in the DP accounting. We fix the randomness of data split using a generator seed of 42. We train the pretrained model using standard SGD with batch size $B = 128$ and learning rate $\eta = 0.1$, and train for $T = 100$ epochs.

Starting from the pretrained checkpoint, we fine-tune the model on D_{priv} using DP-SGD in Opacus with the same implementation adjustments as above, including `ModuleValidator.fix(model)` and setting all ReLU layers to `inplace=False`. We use the SGD optimizer with per-sample gradient clipping norm C, noise multiplier σ, full batch, learning rate 0.1, and train for $T = 20$ epochs. As in the from-scratch experiments, we fix the randomness of data shuffling using a DataLoader generator seed of 42. We generate candidate models by sweeping the learning rate $C \in \{2, 4\}$, and $\sigma \in \{16, 32\}$.

D.3 RS for Merging Checkpoints from the Same Run

We note that the limitation of applying LC to recycled checkpoints is that under DP-SGD and standard privacy accounting, it cannot yield a privacy guarantee better than the least private checkpoint (typically the checkpoint taken at the largest training step). To see why, consider two checkpoints θ_1 and θ_2 taken from the same DP-SGD run at steps $t_1 < t_2$. A linear merge can be written as

$$\theta \;=\; \lambda\theta_1 + (1-\lambda)\theta_2 \;=\; \theta_1 + (1-\lambda)(\theta_2 - \theta_1).$$

Since $\theta_2 - \theta_1$ is exactly the cumulative update from step t_1 to t_2, this merge is essentially equivalent to re-scaling the updates on the segment $t \in \{t_1 + 1, \ldots, t_2\}$ by a factor $(1 - \lambda)$. However, standard DP accounting for DP-SGD is driven by the sampling rate, clipping bound, noise multiplier, and number of steps, and is typically independent of the learning rate. Therefore, such a post-processing does not reduce the privacy cost attributed to the steps up to t_2, and the resulting guarantee cannot be tighter than that of the checkpoint θ_2 itself. In other words, while checkpoint recycling can improve utilities, linear combination over correlated checkpoints is not able to achieve tighter privacy relative to the least private model, without additional assumptions.

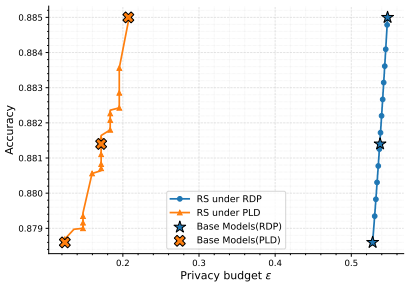
We also consider the checkpoint-merging setting studied in Indri et al. [10], Shejwalkar et al. [20], where the candidate models are checkpoints from a single DP-SGD training run and therefore are generally not independent. This distinction matters: our LC accounting relies on independence noises across the input mechanisms, whereas RS does not. Since RS only requires access to the individual privacy profiles, it remains applicable even when the inputs are correlated checkpoints.

We run experiments under DP-SGD on both an MNIST and CIFAR. We construct three candidate models by taking checkpoints at training steps $T \in \{1, 2, 3\}$ for MNIST and at training steps $T \in \{12, 16, 20\}$ for CIFAR-10 with batch size 256, learning rate 0.1 for MNIST and 0.01 for CIFAR, $\sigma = 0.5$ for MNIST and $\sigma = 0.6$ for CIFAR. The results are shown in Figure 5. We observe that RS still achieves a favorable privacy/utility tradeoff in this correlated-checkpoint regime.

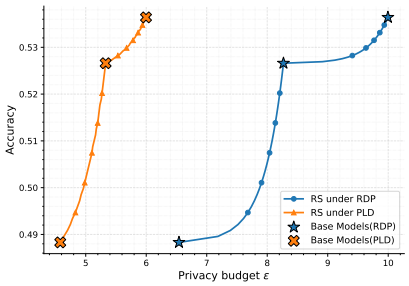
D.4 More Experiment Results

E Additional Experiments

This section reports the additional experiments summarized at the end of Section 8. Unless stated otherwise, the setup follows Appendix D: MNIST uses the logistic regression model and the three

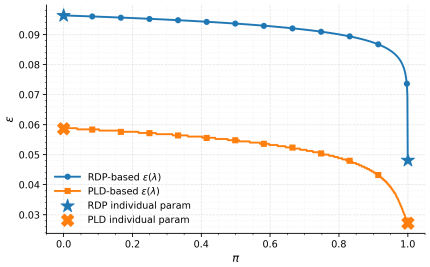


(a) Privacy/utility tradeoffs on MNIST within same run ($\delta = 10^{-5}$)

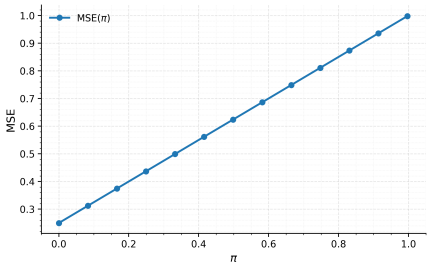


(b) Privacy/utility tradeoffs on CIFAR-10 within same run ($\delta = 10^{-5}$)

Figure 5: Privacy/utility tradeoffs of merging checkpoints from the same run.



(a) DP parameter as π changes



(b) MSE as π changes

Figure 6: Random selection results for mean estimation with $\delta = 10^{-5}$.

input models described there, CIFAR-10 uses the pretraining-based ResNet18 pipeline, and all targets use $\delta' = 10^{-5}$. All numbers are test accuracies in percent.

E.1 Comparison with Retraining and the Best Eligible Input Model

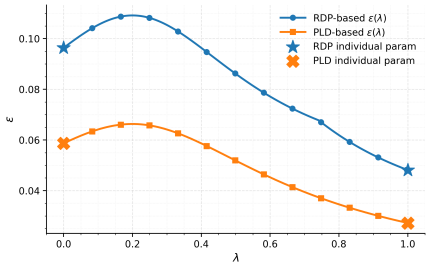
We compare RS and LC with two natural alternatives for meeting a target budget ε' on MNIST under RDP accounting. *Best eligible input model* selects the most accurate input model whose own guarantee already meets ε'. *Retrain* trains a model with DP-SGD from scratch for each target ε', reusing the tuned (C, η, T) of the input models and only recalibrating the noise multiplier σ to hit ε', so that no hyperparameter-search cost is charged to it. Table 1 shows that retraining achieves the accuracy upper bound, as expected, since it retrains on the raw data for each target; both mergers are pointwise no worse than the best eligible input model, and LC recovers most of the gain of retraining without any additional training or access to the data.

Table 1: Comparison with retraining and the best-eligible-input-model baselines on MNIST (RDP accounting, $\delta = 10^{-5}$).

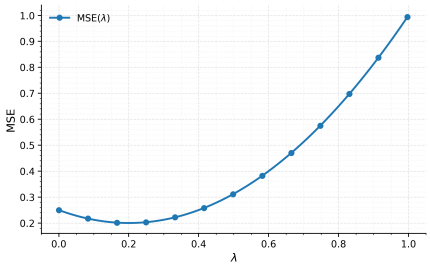
Target ε'	0.55	0.70	0.85	1.00	1.15	1.30
Best eligible input model	88.45	88.45	88.45	88.45	89.12	89.12
Retrain	88.45	88.89	88.99	89.05	89.12	89.26
RS under RDP	88.45	88.45	88.45	88.52	89.12	89.12
LC under RDP	88.45	88.65	88.78	88.91	89.12	89.24

E.2 Non-Private Merging after Joint Release

A further baseline jointly releases all input models and then applies existing non-private merging methods as post-processing: uniform and greedy model soups [23], approximate Fisher-weighted averaging [16], and TIES/DARE-TIES [24, 29]; all input models share a common initialization. Table 2 reports the results. At $\delta = 10^{-5}$, the privacy cost of the joint release under RDP accounting is $\varepsilon = 1.783$ on MNIST and $\varepsilon = 6.560$ on CIFAR-10, both of which exceed the target ranges of our



(a) DP parameter as λ changes



(b) MSE as λ changes

Figure 7: Linear combination results for mean estimation with $\delta = 10^{-5}$.

main experiments (MNIST: 0.55–1.30; CIFAR-10: 2.0–4.0). Moreover, this approach yields a single (privacy, utility) point rather than a privacy/utility curve.

Table 2: Non-private merging of the jointly released input models (RDP accounting, $\delta = 10^{-5}$). The joint release costs $\varepsilon = 1.783$ on MNIST and $\varepsilon = 6.560$ on CIFAR-10.

Dataset	Method	Accuracy (%)
MNIST	Uniform model soup	89.08
MNIST	Greedy model soup	89.12
MNIST	Fisher-weighted averaging	89.03
CIFAR-10	Uniform model soup	87.25
CIFAR-10	Greedy model soup	87.45
CIFAR-10	TIES	87.29
CIFAR-10	DARE-TIES	87.24

E.3 Ablations on the Input Portfolio

Number of input models. We build pools of $N \in \{3, 5, 7\}$ independently trained DP-SGD models on MNIST, where $N = 3$ is exactly the configuration of Appendix D. The added models fill the interior of the original privacy range rather than extending its endpoints, so all pools span the same $[\min_i \varepsilon_i, \max_i \varepsilon_i]$, and any gain reflects a denser, not a wider, portfolio. Table 3 shows that, at every target, accuracies increase with N for both RS and LC. On datasets where the input models differ more substantially in utility, we expect denser pools to matter more, with the same trend.

Table 3: Effect of the number of input models N on MNIST (PLD accounting, $\delta = 10^{-5}$).

Pool	Method	$\varepsilon' = 0.55$	0.70	0.85	1.00	1.15
$N = 3$	RS	88.45	88.45	88.49	88.76	89.12
$N = 3$	LC	88.57	88.72	88.87	88.96	89.21
$N = 5$	RS	88.54	88.89	88.92	89.05	89.12
$N = 5$	LC	88.59	88.96	89.06	89.14	89.23
$N = 7$	RS	88.56	88.93	89.04	89.05	89.12
$N = 7$	LC	88.62	89.05	89.23	89.32	89.35

Distribution of the input privacy levels. We compare a concentrated pool with $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0.62, 0.74, 0.87)$ against a spread-out pool with $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0.49, 0.74, 1.05)$ (Table 4). The two regimes trade privacy coverage for utility. The spread-out pool has a smaller left endpoint and can therefore serve tighter targets: at $\varepsilon' = 0.55$ the concentrated pool admits no feasible merged model at all, while the spread-out pool still attains 88.53%. The advantage of the concentrated pool is precisely its concentration: its input models are closer to each other, so it is easier for the merged model to fully utilize the privacy budget. This effect is more pronounced for LC, which benefits from input models lying close to one another in parameter space.

Table 4: Concentrated versus spread-out input privacy levels on MNIST (PLD accounting, $\delta = 10^{-5}$). "–" indicates that no feasible merged model exists for the target.

Regime	Method	$\varepsilon' = 0.55$	0.70	0.85	1.00	1.15
Concentrated	RS	–	88.93	89.03	89.05	89.05
Concentrated	LC	–	89.02	89.13	89.24	89.24
Spread-out	RS	88.45	88.70	88.99	89.05	89.12
Spread-out	LC	88.53	88.79	89.13	89.20	89.21

E.4 Heterogeneous Input Models

On MNIST, we build a portfolio of three private candidates that differ in architecture, parameter dimension, initialization, learning rate, clipping norm, noise multiplier, and hence privacy level: a DP logistic regression, a DP multilayer perceptron (MLP), and a DP convolutional network (CNN), all at $\delta = 10^{-5}$. LC is not defined for this portfolio, whereas RS remains well-defined and traces a continuous privacy/utility curve (Table 5).

Table 5: Merging a heterogeneous portfolio (logistic regression / MLP / CNN) on MNIST ($\delta = 10^{-5}$). LC is not defined for models with different parameterizations.

Target ε'	0.55	0.70	0.85	1.00	1.15
Best eligible input model	86.98	86.98	86.98	86.98	89.12
RS under RDP	86.98	86.98	87.01	87.25	89.12
RS under PLD	86.99	87.01	87.13	88.07	89.12
LC	N/A	N/A	N/A	N/A	N/A

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: Claims made in the abstract and introduction are supported by algorithms (Section 4), theoretical analysis (Section 5 and 6), and experiments (Section 8).

   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: Section 9

   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: Appendix A-C.

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: We describe experiment datasets, setup, and running details in Section 8 and Appendix D.

   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: We use common open-sourced benchmarks as well as Gaussian synthetic data in the experiments and clearly document them.

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: Section 8 and Appendix D.

   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: [No]

   Justification: Empirical performance gaps are clear so that error bars are not necessary. Also, we do not compare with many baselines, but rather, validate our theoretical claims with empirical evidence.

   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: Appendix D

   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: We confirm that the research in this paper conform with the NeurIPS Code of Ethics.

   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: [N/A]

    Justification: Our paper is on foundational machine learning algorithm design and analysis to preserve privacy. Hence, it does not have direct negative societal impacts.

    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: Our paper poses no such risks of model or data misuse as data are standard common benchmarks and models are standard linear or neural network models.

    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: We properly cited the assets used in the paper.

    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: [Yes]

    Justification: We submit anonymized code.

    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: The paper does not involve 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: The paper does not involve 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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.