END-TO-END DIFFERENTIAL PRIVACY FOR MULTI-STAGE FOUNDATION MODEL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Differential privacy is increasingly used in multi-stage training processes of foundation models, from initial pretraining over different fine-tuning and alignment stages, down to task-specific in-context learning. However, with current privacy accounting, we cannot report an end-to-end privacy guarantee of the final model. The two main obstacles are that (1) the different training stages may protect different privacy units, i.e., they are defined with respect to different neighborhood relations, such as fixed-length token sequences vs. task examples, or documents vs. users, and that (2) the training data overlap between the various stages might be unknown. Since differential privacy guarantees are defined with respect to their own neighborhood relation, individual per stage guarantees cannot be composed directly, and standard composition results require either perfect knowledge of data overlap or pessimistically assuming complete overlap of data between all stages. We close this gap with the first formal framework for end-to-end privacy guarantees under heterogeneous privacy units and unknown but constrained data overlap. Concretely, we formalize each stage as a private mechanism with a stage-specific neighborhood relation and derive tools that give a guarantee for the end-to-end mechanism under generic constraints on data overlap. We show that our bounds are black-box optimal in several cases, and that, surprisingly, converting to a smaller privacy unit, even for DP-SGD, can weaken the guarantee. Turning to practice, we study end-to-end accounting for Gaussian DP mechanisms, showing that our bound is easy to compute even with a large number of stages, and that the most expensive stage(s) dominate the final privacy cost. Our advances enable rigorous end-to-end privacy guarantees for modern multi-stage foundation model training pipelines.

1 INTRODUCTION

Foundation models are commonly trained through a multi-stage pipeline which includes pretraining and diverse adaptation stages, such as fine- and instruction tuning, safety alignment, and task-specific in-context learning. This workflow underlies models from state-of-the-art large language models (LLMs) (Gemini Team, 2023; Jiang et al., 2023; Touvron et al., 2023) to general multi-modal models (Gemini Team, 2025; QwenTeam, 2025; Wei et al., 2025) and has become a standard way to obtain strong performance across domains. At the same time, all stages may use sensitive data and it has been shown that such data can leak through model outputs or model behavior at inference time (Carlini et al., 2021; 2022; 2023; Duan et al., 2024; Nasr et al., 2023). This motivates the need to provide strong privacy guarantees over the full training pipeline.

The standard framework for achieving privacy guarantees is *Differential Privacy* (DP; Dwork et al., 2006b). DP bounds the information that a model can leak about its training data by requiring that models trained on two *neighboring* datasets behave roughly the same. Which datasets count as neighboring is defined by a *neighborhood relationship*, and the element in which two neighboring datasets differ is what the guarantee protects. We refer to this as the *privacy unit*. Choosing a privacy unit, such as a fixed-length token sequence, a labeled example, a document, or all the data of one user, therefore determines what the guarantee means. Recent work has shown that DP can be applied to language model pretraining (McKenna et al., 2025; Sinha et al., 2025) and to downstream adaptations across modalities (Duan et al., 2023; Hong et al., 2024; Li et al., 2022; Liu et al., 2024; Tsai et al., 2025; Wang et al., 2024; Wu et al., 2024; Yu et al., 2022) to yield per-stage privacy guarantees.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

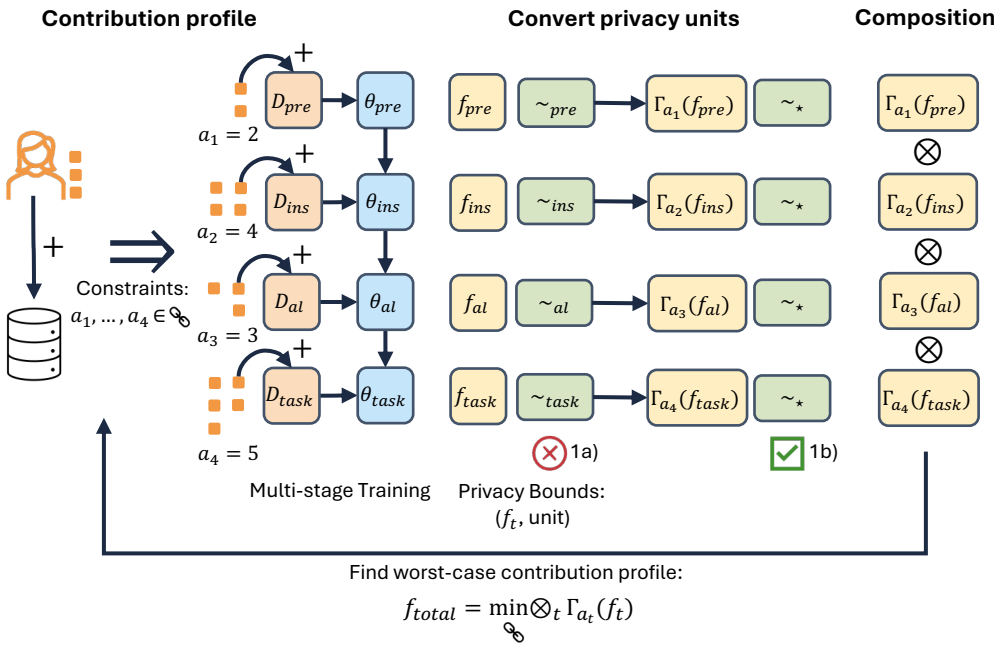


Figure 1: **Overview of end-to-end privacy accounting.** Example of our main theorem in a four-stage training process that contains pretraining (pre), instruction tuning (ins), safety alignment (al), and task-specific finetuning (task). Each stage t has a training dataset D_t and produces a model θ_t under DP. The main obstacle is shown in (1a): each stage has a different privacy unit $\sim_t$, which means the privacy bounds f_t cannot be composed directly. If it is known how the stage-wise datasets overlap, meaning that the contribution profile of one target privacy unit is known, one can convert the stage-wise units to a shared target unit ($f_t \to \Gamma_{a_t}(f_t)$) and compose (1b). Since the exact contribution profile is often not known, but some constraints on it are known, optimizing over all possible profiles is required.

However, those stage-wise guarantees do not, by themselves, determine the privacy guarantee of the final model because of two reasons. (1) Different stages might not protect the same unit. For example, VaultGemma (Sinha et al., 2025) reports sequence-level privacy for sequences of 1024 tokens and McKenna et al. (2025) study sequence-level privacy for 512-token sequences for *pretraining*. In contrast, supervised *adaptation* datasets often have task-specific records, such as labeled examples or documents, as their natural units (Ghazi et al., 2023; Liu et al., 2020; Lowy et al., 2024). Composition theorems, however, apply only to mechanisms that share the same neighborhood relation and privacy unit, so guarantees stated in different units cannot be composed at all. (2) The final guarantee depends on how many stages one target-unit change can affect, and how strongly it affects each of them. This requires knowing how the stage-wise training datasets overlap, which is often not the case in practice. These two obstacles raise the central question of this work:

> **If multiple training stages each satisfy DP under potentially different neighborhood relations, what privacy guarantee does the final model satisfy end-to-end?**

To address this question, we make the following contributions:

1. **Formalizing end-to-end privacy:** We initiate the study of end-to-end privacy guarantees in multi-stage foundation model training pipelines with heterogeneous privacy units and formalize the problem mathematically (Section 3).
2. **End-to-end privacy accounting:** We prove a theorem that upper bounds the end-to-end privacy loss for a target privacy unit under given constraints on the contributions the units can make to each stage (Theorem 4.1), summarized in Figure 1. The upper bound is in the form of an optimization problem that combines group privacy and composition. For the GDP case (Corollary 4.3), we prove a characterization of its solutions that makes finding a solution tractable (Proposition 4.4). If all units can contribute to at most one stage, or can contribute to all stages, the theorem reduces

to a *convert-then-compose* result: first convert stage-wise guarantees to a common unit, then apply standard parallel or sequential composition (Corollary 4.2).

3. **Black-box tightness:** We prove that our bounds cannot be improved by any analysis that treats the stage-wise mechanisms as black boxes (Section 5). In particular, we show that convert-then-compose is black-box optimal when the worst-case privacy unit can be in all of the stage-wise datasets (Theorem 5.1). Theorem 5.4 then shows that a guarantee for a smaller privacy unit, despite being the weaker notion, cannot in general be obtained for free: for clipped DP-SGD-style mechanisms, converting down can weaken the numerical bound, since contributions inside a larger unit may partially cancel before clipping.
4. **Empirical evaluation:** In Section 6, we empirically investigate the GDP case of our theory. We show that it is possible to compute the total privacy cost across a grid of different assumed contribution bounds even with a significant number of training stages (more than would be used for standard training pipelines in practice). We also observe that the total privacy cost is dominated by the stage(s) with the highest privacy cost(s) in the target unit due to the quadratic nature of GDP composition. This means that efforts to reduce the total cost should focus on these stages.

Together, these results give the first formal accounting framework for multi-stage training pipelines whose stages operate with different privacy units and whose data overlap is not fully known. The results potentially also have applications to other settings with multiple heterogeneous data sources, such as multi-agent systems (Patil et al., 2025) and federated learning (Shokri & Shmatikov, 2015).

2 Background and Related Work

2.1 Background on differential privacy (DP), f-DP, and composition

Differential privacy. DP is the standard framework for limiting what can be learned about training data from the output of a training algorithm. Informally, it requires that changing one protected unit in the input dataset has only a limited effect on the distribution of the released output.

Definition 2.1 (Dwork et al., 2006a)**.** *A randomized algorithm $\mathcal{M}\colon \mathcal{D} \to \mathcal{Y}$ is $(\epsilon, \delta, \sim)$-DP if, for all neighboring datasets $D \sim D'$ and all measurable sets $S \subseteq \mathcal{Y}$,*

$$\Pr(\mathcal{M}(D) \in S) \leq e^{\epsilon} \Pr(\mathcal{M}(D') \in S) + \delta. \tag{1}$$

An algorithm is *tightly* $(\epsilon, \delta, \sim)$-DP if the guarantee is not valid for any smaller $\epsilon' < \epsilon$ or $\delta' < \delta$.

Privacy Units. DP is always defined with respect to a *neighborhood relation* $\sim$ on datasets, which specifies the protected *privacy unit*. There are three common ways to interpret "differ in." Under *substitute* adjacency, the two datasets have the same number of units but differ in one unit. Under *add/remove* adjacency, one of D or D' contains one extra privacy unit. *Zero-out* adjacency is similar to add/remove, but the remove operation is changed to a zero-out operation that makes the contribution of a zeroed-out data point to any query zero (Chua et al., 2024c).

For text data, the privacy unit may be a document, a sentence, an (example, label)-pair, or a fixed-length token sequence (McKenna et al., 2025; Sinha et al., 2025) (we defer precise definitions to Appendix A.2). This makes the neighborhood relation central to any end-to-end privacy statement.

f-DP. A single (ϵ, δ)-pair is not enough to optimally characterize composition of algorithms. We therefore work with f-DP (Dong et al., 2022).

Definition 2.2 (Dong et al., 2022)**.** *Let P and Q be probability distributions. The trade-off function $T(P, Q)\colon [0, 1] \to [0, 1]$ maps a false positive rate α to the minimum achievable false negative rate of a test that distinguishes P from Q based on one sample.*

Definition 2.3 (Dong et al., 2022)**.** *A randomized algorithm $\mathcal{M}$ is $(f, \sim)$-DP if, for all neighboring datasets $D \sim D'$, $T(\mathcal{M}(D), \mathcal{M}(D')) \geq f$.*

Definition 2.4 (Dong et al., 2022)**.** *A randomized algorithm $\mathcal{M}$ is $(\mu, \sim)$-Gaussian DP (GDP) if it is G_μ-DP for $G_\mu(\alpha) = \Phi(\Phi^{-1}(1 - \alpha) - \mu)$, where Φ is the standard Gaussian CDF.*

Composition. f-DP admits clean and tight composition theorems: when two mechanisms are applied to the same dataset, the composition is *sequential*. When they are applied to disjoint datasets,

the composition is *parallel*. In both cases, the composition may be *adaptive*, meaning that later mechanisms may depend on earlier outputs.

For trade-off functions f_1, f_2, we write $f_1 \otimes f_2$ for their f-DP sequential-composition operator, and $\bigotimes_{t=1}^{T} f_t$ for repeated sequential composition. See Dong et al. (2022) for the precise definition of the $\otimes$ operation.

Theorem 2.5 (Dong et al., 2022)**.** *The adaptive sequential composition of an $(f_1, \sim)$-DP algorithm with an $(f_2, \sim)$-DP algorithm is $(f_1 \otimes f_2, \sim)$-DP. This bound is tight.*

Theorem 2.6 (Smith et al., 2022, Theorem 5)**.** *The adaptive parallel composition of an $(f_1, \sim)$-DP algorithm with an $(f_2, \sim)$-DP algorithm on disjoint datasets is $(\mathrm{lce}(f_1, f_2), \sim)$-DP, where* lce *denotes the lower convex envelope.*

The lower convex envelope is essentially a minimum, but ensures that the result is a trade-off function. We defer the precise definition to Appendix A.1.

In practice, tight composition can be approximated numerically (Doroshenko et al., 2022; Gomez et al., 2026; Kulynych et al., 2024) using so called *privacy loss distributions* (PLD; Koskela et al., 2020). Trade-off curves and PLDs are essentially equivalent: it is possible to losslessly convert between them (Zhu et al., 2022).

Group privacy. DP bounds the change that can occur when one privacy unit is changed. Group privacy results generalize these guarantees to changes of groups of privacy units. For functions g, we write $g^{\circ k}$ for k-fold functional composition. We also define $\Gamma_k(f) := 1 - (1 - f)^{\circ k}$, with $\Gamma_0(f)(\alpha) = 1 - \alpha$.

Theorem 2.7 (Dong et al., 2022)**.** *If a mechanism is $(f, \sim)$-DP, then it is $(\Gamma_k(f), \sim)$-DP for groups of size k. If a mechanism is $(\mu, \sim)$-GDP, then it is $(k\mu, \sim)$-GDP for groups of size k.*

2.2 Related Work

Privacy in Language Models. Privacy risks in language models have been documented through memorization and extraction attacks (Carlini et al., 2021) and discussed more broadly as a mismatch between data protection mechanisms and the contextual nature of privacy in language (Brown et al., 2022). In particular, Brown et al. (2022) argue that privacy for language models is broader than standard data protection notions, because privacy in natural language is contextual and may not align with well defined records. Our work addresses a narrower but practically important question within formal DP: how to obtain an end to end guarantee once explicit privacy units have been chosen for multiple training stages.

DP Training. Recent work has studied DP for language model pretraining and for downstream adaptation. On the pretraining side, large language models have been trained with sequence-level DP (McKenna et al., 2025; Sinha et al., 2025). On the adaptation side, prior work develops DP fine-tuning and prompting methods for language models which usually operate on the level of individual labeled samples or individual documents (Duan et al., 2023; Hong et al., 2024; Li et al., 2022; Wu et al., 2024; Yu et al., 2022). Both lines of work analyze the stages separately. As a result, they do not provide an end-to-end privacy guarantee for the final model when privacy losses from all stages must be composed, especially if the stages use different privacy units. Our contribution is complementary: rather than proposing a new private training method, we study how to formally compose stage-wise DP guarantees across multi-stage pipeline.

User-level DP. Works on user-level DP (Ghazi et al., 2023; Liu et al., 2020) also study different privacy units, typically example-level vs. user-level, or element-level (Asi et al., 2019). Ghazi et al. (2023) prove a conversion theorem from example-level to user-level DP, similar to the one-stage special case of our main theorem. Chua et al. (2024a) evaluate user-level language model finetuning techniques.

Maximum Overlap Composition. Smith et al. (2022) prove an f-DP composition theorem which is similar to our in that they optimize over worst-case overlap of several queries. However, they do not consider heterogeneous privacy units, and only consider predicate queries on discrete data.

3 Problem Setup and Privacy Units

We consider a multi-stage training pipeline with T stages, with a dataset D_i, $1 \leq i \leq T$ per stage. The first stage produces a model $\theta_1 = \mathcal{M}_1(D_1)$, and subsequent stages output models $\theta_i = \mathcal{M}_i(D_i, \theta_1, \ldots, \theta_{i-1})$. Thus, the final released mechanism is the adaptive composition $\mathcal{M}(D_1, \ldots, D_T) := (\theta_1, \ldots, \theta_T)$ releasing all models. We assume that each stage has a privacy guarantee with neighborhood relation $\sim_i$. Our goal is to characterize the privacy of the final released model with respect to a target neighborhood relation $\sim_\star$, which specifies the *target privacy unit* of the final end-to-end claim.

The neighborhood relations $\sim_i$ may differ, because each stage may protect a different *privacy unit*. We use the term *privacy unit* to refer to the object changed by the neighborhood relation. For example, pretraining typically uses fixed-length sequences as the privacy unit (McKenna et al., 2025; Sinha et al., 2025), while supervised finetuning would naturally use (example, label)-pairs as the unit. Note that each privacy unit is paired with an adjacency notion, such as add/remove, substitute, and zero-out adjacency, which can also differ between stages. For formal examples, see the definitions for text-based neighborhood relations in Appendix A.2.

To convert between neighborhood relations, we need to characterize how many units of one neighborhood correspond to a unit in the other. A neighborhood $\sim$ can be viewed as defining a graph whose vertices are datasets and whose edges connect neighboring datasets. For two datasets D and D', let $d_\sim(D \to D')$ be the shortest-path distance from D to D' in this graph. If no path exists, we set $d_\sim(D \to D') = \infty$. For a base relation $\sim$ and a target relation $\sim_\star$, define

$$k(\sim \to \sim_\star) := \sup_{D \sim_\star D'} d_\sim(D \to D'). \tag{2}$$

Thus, $k(\sim \to \sim_\star)$ is the largest number of base-neighboring changes needed to realize one target-neighboring change. If this quantity is finite, then every target-neighboring pair is a group of at most $k(\sim \to \sim_\star)$ base-neighboring changes.

Example 1: Sequence-level Neighborhood

Consider a fixed tokenized corpus, and define two sequence-level neighborhood relations on it: $\sim$ changes one aligned 512-token chunk, while $\sim_\star$ changes one aligned 1024-token chunk. If both relations use zero-out adjacency, then $k(\sim \to \sim_\star) = 2$, because one 1024-token target-unit change corresponds to two 512-token base-unit changes. The chunks being aligned means that every 1024-token chunk contains exactly two 512-token chunks. We provide comprehensive tables with the value of $k(\sim \to \sim_\star)$ for different sequence lengths in Appendix B.

For an end-to-end guarantee, the relation between the datasets D_i also matters. The simplest case is when it is fully known in which training stage's dataset a privacy unit can appear. If each unit only affects one stage, parallel composition can be used. In contrast, if the target unit may affect all stages, the losses compose sequentially. Intermediate cases where one unit can affect some but not all stages can be handled with combination of parallel and sequential composition.

The harder case, which we will deal with, is when the possible contributions are not fully known. If nothing about the contributions is known, the case collapses to the worst-case of sequential composition between all stages. However, if some constraints on the contributions can be assumed, the result is a non-trivial optimization problem, which we explore in the next section.

4 Composition Under Heterogeneous Privacy Units and Constrained Data Overlap

In this section, we provide a general theorem for end-to-end privacy accounting multi-stage pipelines under unknown, but constrained overlap between each stage-wise dataset. First we give a general theorem for f-DP, and then state the GDP version as a corollary. Then we show that the GDP version has structure which enables a practical algorithm for finding the solution.

Let u denote one target privacy unit under a target neighborhood $\sim_\star$, such as a user or document, and let $a_t(u)$ be the number of stage-t training units from neighborhood $\sim_t$ contributed by u. For example, if stage t trains on sequences, then $a_t(u)$ is the number of sequences associated with u; if it trains on examples, then $a_t(u)$ is the number of examples associated with u. To simplify notation, we will assume in this section that the target privacy unit is larger than any of the stage-wise units and that all are using the same adjacency, so we have $k(\sim_t \to \sim_\star) = \sup_u a_t(u)$.

We do not know the values $a_t(u)$ exactly, but assume the following bounds are known: for all u,

$$0 \le a_t(u) \le A_t \quad \forall t, \qquad \sum_{t=1}^{T} \mathbf{1}\{a_t(u) > 0\} \le r, \qquad \sum_{t=1}^{T} a_t(u) \le B. \tag{3}$$

Here, A_t is the contribution cap at stage t, r bounds the number of stages in which any one user or document may occur, and B bounds its total contribution across the full pipeline. The bounds may for example be derived from organizational policies or data provenance information.[1] If exact bounds are not known, it is still possible to compute the results for all plausible bounds to understand how sensitive the privacy guarantee is to the precise values, which we will do in Section 6.

Theorem 4.1. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be algorithms that are respectively $(f_t, \sim_t)$-DP. Let $\sim_\star$ be a target neighborhood relation. Under the assumptions in Equation (3), the composition of $\mathcal{M}_1, \ldots, \mathcal{M}_T$ is $(f_{\text{final}}, \sim_\star)$-DP with*

$$f_{\text{final}} := \operatorname*{lce}_{\substack{(a_1,\ldots,a_T)\in\mathbb{Z}_{\ge 0}^T \\ 0\le a_t\le A_t,\ \sum_{t=1}^T \mathbf{1}\{a_t>0\}\le r,\ \sum_{t=1}^T a_t\le B}} \bigotimes_{t=1}^{T} \Gamma_{a_t}(f_t). \tag{4}$$

Theorem intuition. The end-to-end privacy guarantee under constrained stage-wise contributions is a sequential composition over the stages for the worst-case privacy unit that satisfies the constraints.

Proof sketch. The proof starts with the observation that group privacy and sequential composition hold for individual privacy units, though they are stated for worst-case units. The objective and constraints only depend on the unit through $a_t(u)$, so finding the worst-case unit is equivalent to finding an integer vector $(a_1, \ldots, a_T)$ that satisfies the constraints and maximizes privacy loss. See Appendix C for the full proof. □

Equation (4) generalizes standard composition results. Setting $r = \infty$, $B = \infty$, $A_t = 1$ for all t recovers sequential composition, while changing $r = 1$ recovers parallel composition.

If it is known that the stage-wise datasets have no overlap ($r = 1, B = \infty$), or it is pessimistically assumed they fully overlap ($r = B = \infty$), Theorem 4.1 produces a convert-then-compose result, where each stage-wise privacy guarantee is first converted to a common neighborhood relation $\sim_\star$ and then parallel or sequential composition is used.

Corollary 4.2 (Convert-then-compose)**.** *In the setting of Theorem 4.1 with $B = \infty$, when $r = 1$, $f_{\text{final}} = \operatorname{lce}_{1\le t\le T} \Gamma_{A_t}(f_t)$, and when $r = \infty$, $f_{\text{final}} = \bigotimes_{t=1}^{T} \Gamma_{A_t}(f_t)$.*

Proof. Follows immediately from Theorem 4.1. □

This provides an easy to use privacy bound for cases without overlap or full overlap.

Next, we state the GDP special case, which is amenable to numerical optimization for practical values of T, B and r, which is needed for applications.

Corollary 4.3. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be algorithms that are respectively $(\mu_t, \sim_t)$-GDP. Let $\sim_\star$ be a target neighborhood relation. Under the assumptions in Equation (3), the composition of $\mathcal{M}_1, \ldots, \mathcal{M}_T$ is $(\mu_{\text{final}}, \sim_\star)$-GDP with*

$$\mu_{\text{final}}^2 := \max_{\substack{(a_1,\ldots,a_T)\in\mathbb{Z}_{\ge 0}^T \\ 0\le a_t\le A_t,\ \sum_{t=1}^T \mathbf{1}\{a_t>0\}\le r,\ \sum_{t=1}^T a_t\le B}} \sum_{t=1}^{T} a_t^2 \mu_t^2. \tag{5}$$

[1] Knowledge of this type can correspond to stronger constraints than what we assume. For example, a hospital may know that their data could only be in the stages they trained. But with many data sources with different constraints, the optimization problem quickly becomes very complicated, while our weaker constraints make the problem tractable in the GDP case (Proposition 4.4).

Proof sketch. The result follows from properties of GDP (Dong et al., 2022) and Theorem 4.1. See Appendix C for details. □

The following result characterizes solutions to Equation (5).

Proposition 4.4. *When $\mu_t > 0$ for all $t \in \{1, \dots T\}$, any optimal solution $(a_1^\star, \dots, a_T^\star)$ to Equation (5) has all but at most one nonzero coordinate at their stage-wise caps. That is, for all but at most one stage t, $a_t^\star \in \{0, A_t\}$.*

Proof sketch. The proof shows that if a solution had two non-zero coordinates that are not at their caps, moving one contribution from one coordinate to the other would increase the objective. See Appendix C for details. □

Using this characterization, it is clear that we can find the optimal solution by enumerating all the subsets of stages set to their caps, and additionally enumerating over the remaining stages for the at most one stage that is not at the cap. We present the exact procedure in Algorithm 1 in Appendix C. The algorithm has running time $O\Big(T \sum_{s=0}^{r} \binom{T}{s}\Big)$, which is feasible when the number of stages T or the overlap bound r is small, as is often the case in practice.

5 Black-box Optimality Results

In this section, we present several results showing where Theorem 4.1 is tight. We also study the limits of converting to smaller privacy units.

Convert-then-compose optimality. One might ask whether treating the unit conversion and composition separately, as in Theorem 4.1 and Corollary 4.2, is optimal, or could there be a process that cannot be decomposed and certifies stronger privacy bounds than convert-then-compose? The following informal theorem shows that convert-then-compose is indeed optimal if both the conversion and composition are optimal, and the worst-case privacy unit can be in all stages, meaning $r = \infty$ and $B = \infty$ in Theorem 4.1.

Theorem 5.1 (Informal)**.** *Convert-then-compose is optimal for f-DP when both the conversion and composition are optimal and the worst-case privacy unit can be in all stages.*

Proof sketch. We show that if the combination of conversion and composition satisfied a stronger f-DP bound than given by convert-then-compose, one of the conversion or composition would have to be suboptimal. See Proposition D.2 in the Appendix for the full theorem and proof. □

Next, we present the $T = 1$ special case of Theorem 4.1, which only converts between neighborhood relations. For completeness, we present a more general form that includes GDP and (ϵ, δ)-DP, and allows conversions from larger to smaller units.

Theorem 5.2 (Privacy-unit conversion)**.** *Let $\mathcal{M}$ be a randomized algorithm satisfying DP with respect to a base neighborhood relation $\sim$. Let $\sim_\star$ be a target neighborhood relation, and suppose $K := k(\sim \to \sim_\star) < \infty$. Then the following hold.*

1. *If $\mathcal{M}$ is $(f, \sim)$-DP, then $\mathcal{M}$ is $(f_\star, \sim_\star)$-DP with $f_\star = \Gamma_K(f)$.*
2. *If $\mathcal{M}$ is $(\mu, \sim)$-GDP, then $\mathcal{M}$ is $(K\mu, \sim_\star)$-GDP.*
3. *If $\mathcal{M}$ is $(\epsilon, \delta, \sim)$-DP, then $\mathcal{M}$ is $(K\epsilon, \delta_\star, \sim_\star)$-DP with $\delta_\star = \delta \sum_{j=0}^{K-1} e^{j\epsilon}$.*

Theorem intuition. A change in the target privacy unit may correspond to several changes in the unit used by the stage-wise DP guarantee. If one target-unit change can always be decomposed into at most K base-unit changes, then the base-unit guarantee can be lifted to the target unit by group privacy with group size K.

Proof. If $K = 1$, then $D \sim_\star D'$ implies $D \sim D'$, so the claim follows from monotonicity of DP under restriction of the neighboring pairs, see Lemma D.3 in Appendix D.2. If $K \geq 2$, any $\sim_\star$-neighboring pair can be connected by at most K $\sim$-neighboring steps, so the claims are exactly the standard group-privacy bounds. The f-DP and GDP forms follow from Theorem 2.7, and the (ϵ, δ) form follows from Vadhan (2017, Lemma 7.2.2). □

Conversion tightness. Theorem 5.2 is a black-box conversion: it uses only the stage-wise DP guarantee and the relationship between the base and target neighborhoods. We show that several of these bounds are tight in the black-box setting. We present informal versions here, and defer the full theorems and proofs to Appendix D.3.

Theorem 5.3 (Informal)**.** *The f-DP and GDP bounds in Theorem 5.2 cannot be improved in general when $K \geq 2$.*

Proof sketch. The proof sets $\sim$ to be example-level privacy, $\sim_\star$ to be privacy for K-sized groups, both under any of substitute, add/remove or zero-out. We then show that the Gaussian mechanism satisfies the bounds of Theorem 5.2 tightly. The full theorem statement and proof are given in Theorem D.4 in the Appendix. □

The following informal theorem presents a surprising case about conversion under zero-out. It shows that converting a larger unit to a smaller one requires $K > 1$, meaning the numerical privacy bound increases, even though a smaller privacy unit is seemingly weaker. Our counterexample follows the standard privacy accounting assumptions of DP-SGD, meaning that this phenomenon is not restricted to pathological mechanisms.

Theorem 5.4 (Informal)**.** *Larger-to-smaller conversions under zero-out must have $K > 1$ in general, even under standard assumptions for clipped DP-SGD.*

Proof sketch. The proof constructs a query for which the clipped value changes by more under a partial zero-out than a complete zero-out. This is done by dividing the query into two halves which partially cancel each other, allowing the partial zero-out to flip the sign of the query while a complete zero-out cannot do so. The full theorem and proof are in Theorem D.5 in the Appendix. □

6 Experiments

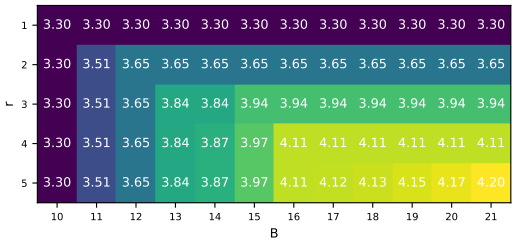
Empirical Research Questions. In this section, we aim to empirically answer the following questions:

- **Q1**: The quadratic objective in Corollary 4.3 suggests that the largest product $A_t\mu_t$ dominates μ_{final}. How significant is this effect with plausible values of $A_{1:T}$ and $\mu_{1:T}$?
- **Q2**: Is solving the optimization problem in Corollary 4.3 with Algorithm 1 practical across a grid of relevant B and r values for typical values of T?

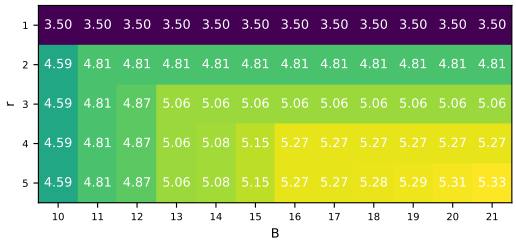
Setup. We investigate both questions in Figure 2. We set $A_{1:T}$ and $\mu_{1:T}$ from Corollary 4.3 to plausible values: the stage-wise contribution caps $A_{1:T} = (10, 5, 3, 2, 1)$ are a decreasing sequence. This is reflecting the fact that early-stage datasets, especially pretraining, are often large, and there is little information about them, so they may need larger caps than later stages. We initially set $\mu_{1:T} = (0.33, 0.17, 0.49, 0.78, 1.2)$ in Figure 2 (a). $\mu_1 = 0.33$ is chosen to be close to the published $(\epsilon \leq 2, \delta \leq 1.1 \cdot 10^{-10})$-DP guarantee for VaultGemma (Sinha et al., 2025). The subsequent μ_t values are chosen to approximate $\epsilon_t \in (1, 3, 5, 8)$, which are often used in the DP finetuning literature (Duan et al., 2023), at $\delta = 1.1 \cdot 10^{-10}$.

Q1. The main finding from Figure 2 (a) is that the total privacy cost is dominated by the largest stage-wise privacy cost $A_t\mu_t$, accounting for both the base cost μ_t and the contribution cap A_t. The largest product in this case is with $t = 1$. As a result, the difference between parallel composition $(r = 1, B = 10, \mu_{\text{final}} = 3.3)$ and sequential composition $(r = 5, B = 21, \mu_{\text{final}} = 4.2)$ is smaller than what might be expected. In panels (b) and (c), we change $\mu_{1:T}$ to have multiple approximately equal $A_t\mu_t$ products: two in (b) and all five in (c). We observe that r becomes more important than B, and the difference between parallel and sequential composition grows in both cases. This means that in cases with a single dominant stage-wise privacy cost, efforts to reduce the total privacy cost should focus on that one stage. The μ_t for that stage can be reduced for example by selecting another checkpoint with smaller privacy cost if one is available. A_t can be reduced by limiting the contributions a single privacy unit can make, or obtaining better information on how many contributions each privacy unit can make in cases where A_t is a conservative upper bound.

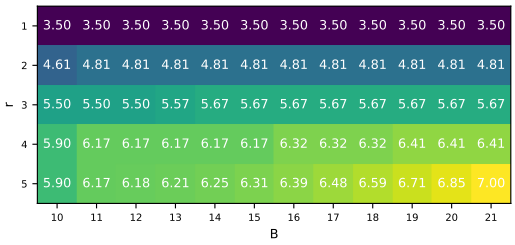
We also investigate a decreasing and a constant μ budget in Figure 3 in the Appendix. Both cases are dominated by the first stage with the largest total cost, especially the decreasing μ case.



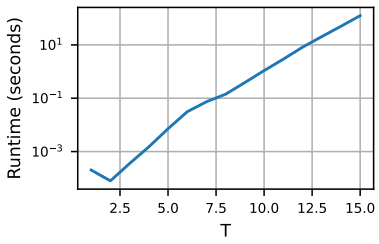
(a) $\mu_{1:T} = (0.33, 0.17, 0.49, 0.78, 1.2)$



(b) $\mu_{1:T} = (0.33, 0.17, 0.49, 0.78, 3.5)$



(c) $\mu_{1:T} = (0.33, 0.6, 1.0, 1.4, 3.5)$



(d) Runtime for whole grid.

Figure 2: (a) to (c): Total μ_{final} from Corollary 4.3 for $A_{1:T} = (10, 5, 3, 2, 1)$ and different values of $\mu_{1:T}$, with $T = 5$ stages. In (a), $\mu_{1:T}$ was chosen to reflect a plausible training process, where r and B are seen to be equally important since $A_1\mu_1$ is larger than $A_t\mu_t$ for other indices. Panels (b) and (c) demonstrate scenarios without a single dominant $A_t\mu_t$: in (b) there are two dominant products, and in (c) all products are approximately equal. We can see that r becomes more important in these scenarios. Panel (d) shows the total time it takes to produce the entire grid as in (a)-(c) with Algorithm 1 for different numbers of stages T. Here $A_{1:T} = (T, \dots, 1)$, $\mu_t = 0.1$ for all t, r ranges from 1 to T and B ranges from $\max A_t$ to $\sum_{i=1}^{T} A_t$.

Q2. In Figure 2 (d), we plot the time it took to compute the whole grid as in panels (a)-(c) with Algorithm 1 for different values of T. The time increases approximately exponentially, but computing the whole grid is still practical with $T = 15$, which would be a very large number of stages in practice. Our implementation is basic: it is written in Python, not heavily optimized, runs on a single core, and was run on a M3 MacBook Pro laptop. Despite this, it covers all practical pipelines in modern foundation model training.

7 Discussion and Conclusions

We studied DP for multi-stage training pipelines, observing that stage-wise privacy guarantees do not by themselves determine the privacy guarantee of the final released model. Instead, end-to-end privacy depends crucially on the protected privacy unit(s) and the data overlap across stages. To address this, we started by mathematically formalizing end-to-end privacy accounting for multi-stage pipelines with heterogeneous privacy units. We proved an upper bound (Theorem 4.1) on the privacy loss for a target privacy unit, with constraints on the contributions that one unit can make at each stage. We then proved that our upper bound is blackbox-tight in several cases, and proved a surprising result that converting a larger privacy unit to a smaller unit can weaken privacy, even for DP-SGD-style mechanisms. We studied the GDP case of our upper bound empirically, and found that the stage(s) with the largest privacy cost in the target unit dominate the end-to-end cost, so those stage(s) should be focused on when minimizing the privacy cost. We also demonstrated that the GDP upper bound can easily be computed with up to 15 stages across a grid of assumed constraint bounds. Our results open the door for formal privacy accounting of modern multi-stage training pipelines that have heterogeneous privacy units and unknown data overlap across stages. A valuable direction for future work would be developing numerical group privacy accounting that works with tradeoff functions directly, or implements the same operation for privacy loss distributions. This would enable applying Theorem 4.1 directly in cases where the optimization problem in the upper bound is tractable, without going through GDP.

AI USE STATEMENT

We used generative AI tools to come up with initial versions of Corollary 4.3 and Proposition 4.4. These were refined by the authors. The f-DP generalization of Corollary 4.3 in Theorem 4.1 is from the authors. The AI tools also came up with an initial proof for Proposition 4.4, which required minor corrections from the authors. All other proofs were produced by the authors. We have not used generative AI to design or provide feedback on research methodology or experiments, implement methods or interpret results. Generating synthetic data sets, assisting with translation, cleaning and reformatting dataset and supporting qualitative and thematic data analysis are not applicable to this work. We have also used generative AI tools to assist writing the experiment code and improve the writing of the paper. We have reviewed all AI-generated mathematics and corrected that as needed. All results in the paper are novel to the best knowledge of the authors, except when stated otherwise in the text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The experiment setup is discussed in Section 6. We have provided the code as supplementary material on OpenReview.

REFERENCES

Hilal Asi, John Duchi, and Omid Javidbakht. Element Level Differential Privacy: The Right Granularity of Privacy, 2019. URL https://arxiv.org/abs/1912.04042.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pp. 2280–2292, 2022.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270. USENIX Association, 2023.

Lynn Chua, Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Daogao Liu, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Mind the Privacy Unit! User-Level Differential Privacy for Language Model Fine-Tuning. In *First Conference on Language Modeling*, 2024a.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How Private are DP-SGD Implementations? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8904–8918. PMLR, 2024b.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Scalable DP-SGD: Shuffling vs. Poisson Subsampling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, pp. 70026–70047, 2024c.

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.

Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. In *Proceedings on Privacy Enhancing Technologies*, volume 4, pp. 552–570, 2022.

Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. On the privacy risk of in-context learning, 2024. URL https://arxiv.org/abs/2411.10512.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006b.

Gemini Team. Gemini: a family of highly capable multimodal models, 2023. URL https://arxiv.org/abs/2312.11805.

Gemini Team. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities., 2025. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf. Accessed 2026-05-01.

Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Raghu Meka, and Chiyuan Zhang. User-Level Differential Privacy With Few Examples Per User. In *Advances in Neural Information Processing Systems*, volume 36, pp. 19263–19290, 2023.

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

Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. DP-OPT: Make large language model your differentially-private prompt engineer. In *The Twelfth International Conference on Learning Representations*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing Tight Differential Privacy Guarantees Using FFT. In *The 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2560–2569. PMLR, 2020.

Bogdan Kulynych, Juan Felipe Gomez, Georgios Kaissis, Flavio Calmon, and Carmela Troncoso. Attack-Aware Noise Calibration for Differential Privacy. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, pp. 134868–134901, 2024.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022.

Jing Liu, Andrew Lowy, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. Efficient differentially private fine-tuning of diffusion models, 2024. URL https://arxiv.org/abs/2406.05257.

Yuhan Liu, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: User vs item-level privacy. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20965–20976, 2020.

Andrew Lowy, Daogao Liu, and Hilal Asi. Faster algorithms for user-level private stochastic convex optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 96071–96100, 2024.

Ryan McKenna, Yangsibo Huang, Amer Sinha, Borja Balle, Zachary Charles, Christopher A. Choquette-Choo, Badih Ghazi, Georgios Kaissis, Ravi Kumar, Ruibo Liu, Da Yu, and Chiyuan Zhang. Scaling Laws for Differentially Private Language Models. In *Forty-Second International Conference on Machine Learning*, 2025.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable Extraction of Training Data from (Production) Language Models, 2023. URL https://arxiv.org/abs/2311.17035.

Vaidehi Patil, Elias Stengel-Eskin, and Mohit Bansal. The Sum Leaks More Than Its Parts: Compositional Privacy Risks and Mitigations in Multi-Agent Collaboration, 2025. URL https://arxiv.org/abs/2509.14284.

QwenTeam. Qwen3-livetranslate: Real-time multimodal interpretation — see it, hear it, speak it, 2025. URL https://qwen.ai/blog?id=0ab294b059db3c8704200e57d5e4246f27abb6a6. Accessed 2026-05-01.

Reza Shokri and Vitaly Shmatikov. Privacy-Preserving Deep Learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pp. 1310–1321. Association for Computing Machinery, 2015.

Amer Sinha, Thomas Mesnard, Ryan McKenna, Daogao Liu, Christopher A. Choquette-Choo, Yangsibo Huang, Da Yu, George Kaissis, Zachary Charles, Ruibo Liu, Lynn Chua, Pritish Kamath, Pasin Manurangsi, Steve He, Chiyuan Zhang, Badih Ghazi, Borja De Balle Pigem, Prem Eruvbetine, Tris Warkentin, Armand Joulin, and Ravi Kumar. VaultGemma: A differentially private Gemma model, 2025. URL https://arxiv.org/abs/2510.15001.

Josh Smith, Hassan Jameel Asghar, Gianpaolo Gioiosa, Sirine Mrabet, Serge Gaspers, and Paul Tyler. Making the Most of Parallel Composition in Differential Privacy. *Proceedings on Privacy Enhancing Technologies*, 2022(1):253–273, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Yu-Lin Tsai, Yizhe Li, Chia-Mu Yu, Xuebin Ren, Po-Yu Chen, Zekai Chen, and Francois Buet-Golfouse. Differentially private fine-tuning of diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4561–4571. IEEE, 2025.

Salil P. Vadhan. The Complexity of Differential Privacy. In Yehuda Lindell (ed.), *Tutorials on the Foundations of Cryptography*, pp. 347–450. Springer International Publishing, 2017.

Haichen Wang, Shuchao Pang, Zhigang Lu, Yihang Rao, Yongbin Zhou, and Minhui Xue. dp-promise: Differentially private diffusion probabilistic models for image synthesis. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1063–1080, 2024.

Yake Wei, Yu Miao, Dongzhan Zhou, and Di Hu. Moka: Multimodal low-rank adaptation for MLLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4782–4817. PMLR, 2022.

A Additional Background

A.1 Lower Convex Envelope

Definition A.1 (Smith et al., 2022)**.** *Let f_1 and f_2 be trade-off functions. The lower convex envelope* $\mathrm{lce}(f_1, f_2)$ *is*

$$\mathrm{lce}(f_1, f_2)(\alpha) = \sup\{f(\alpha) \mid f \text{ is convex and } f \leq \min\{f_1, f_2\}\}. \tag{6}$$

Lemma A.2 (Smith et al., 2022)**.** *The lower convex envelope of two trade-off functions is a trade-off function.*

A.2 Formal text neighborhood relations

Zero-out neighborhood details The zero-out neighborhood relation (Chua et al., 2024b) is a version of the standard add/remove neighborhood where neighboring datasets have the same size. Specifically, two datasets are zero-out neighbors if they have the same number of points, only differ in one index, and one of the differing datapoints is a zero-out marker $\perp$. The marker $\perp$ is a formal value that makes any query q return 0. To make conversion to zero-out neighborhoods from add/remove or substitute possible, we allow them to add or substitute $\perp$.

Sequence-level neighborhoods Let D denote a text dataset. We assume there is a tokenizer Tok that splits a dataset into a list of tokens and a sequencer Seq_L that combines the tokens of $\mathrm{Tok}(D)$ into L-length sequences $\mathrm{Seq}_L(\mathrm{Tok}(D)) = t_1, \ldots, t_{n_t}$, where t_i is a list of L tokens. Seq can truncate tokens or add padding tokens to ensure every sequence has exactly L tokens.

Definition A.3. *Datasets D and D' are L-token sequence-level zero-out neighbors, denoted $D \sim_{t,L}^{ZO} D'$, if the sequence lists $t_1, \ldots, t_{n_t} = \mathrm{Seq}_L(\mathrm{Tok}(D))$ and $t'_1, \ldots, t'_{n'_t} = \mathrm{Seq}_L(\mathrm{Tok}(D'))$ have $n_t = n'_t$, they only differ in one index k^*, and either $t_{k^*} = \perp$ or $t'_{k^*} = \perp$, where $\perp$ is a zero marker that makes any query return 0.*

Definition A.4. *Datasets D and D' are L-token sequence-level add/remove neighbors, denoted $D \sim_{t,L}^{A/R} D'$, if the sequence lists $t_1, \ldots, t_{n_t} = \mathrm{Seq}_L(\mathrm{Tok}(D))$ and $t'_1, \ldots, t'_{n'_t} = \mathrm{Seq}_L(\mathrm{Tok}(D'))$ only differ by the addition or removal of one sequence with length $\leq L$.*

Note that we allow adding a sequence shorter than the length to allow converting a neighborhood with a longer sequence to a neighborhood with a shorter one (see Theorem B.1). We still require that this addition or removal does not change any of the other sequences, which may be difficult to implement in practice. This makes zero-out neighborhood easier to use than add/remove.

Definition A.5. *Datasets D and D' are L-token sequence-level substitute neighbors, denoted $D \sim_{t,L}^{S} D'$, if the sequence lists $t_1, \ldots, t_{n_t} = \mathrm{Seq}_L(\mathrm{Tok}(D))$ and $t'_1, \ldots, t'_{n'_t} = \mathrm{Seq}_L(\mathrm{Tok}(D'))$ have $n_t = n'_t$ and they only differ in one index k^*.*

We will consider sequence-level neighborhoods for different lengths L_1 and L_2. In these cases, we will assume the longer length is divisible by the shorter, and that the sequences are aligned such that any shorter sequence overlaps to exactly one longer sequence, unless otherwise stated.

Note that the sequencer is an integral part of the definitions, so if two training stages do not use aligned sequences, they have different neighborhoods.

Example-level neighborhoods Now we assume the dataset D consists of pairs (x_i, y_i) for $1 \leq i \leq n$, where each x_i is an example string of text, and each y_i is an associated label. For an other dataset D', we define (x'_i, y'_i) and n' in the same way.

Definition A.6. *Datasets D and D' are example-level zero-out neighbors, denoted $D \sim_{e}^{ZO} D'$, if the datasets $D = ((x_1, y_1), \ldots, (x_n, y_n))$ and $D' = ((x'_1, y'_1), \ldots, (x'_{n'}, y'_{n'}))$ have $n = n'$, they only differ in one index k^*, and either $(x_{k^*}, y_{k^*}) = \perp$ or $(x'_{k^*}, y'_{k^*}) = \perp$, where $\perp$ is a zero marker that makes any query return 0.*

Definition A.7. *Datasets D and D' are example-level add/remove neighbors, denoted $D \sim_{e}^{A/R} D'$, if the datasets $D = ((x_1, y_1), \ldots, (x_n, y_n))$ and $D' = ((x'_1, y'_1), \ldots, (x'_{n'}, y'_{n'}))$ only differ by the addition or removal of one (x, y)-pair.*

Definition A.8. *Datasets D and D' are example-level substitute neighbors, denoted $D \sim_e^S D'$, if the datasets $D = ((x_1, y_1), \ldots, (x_n, y_n))$ and $D' = ((x'_1, y'_1), \ldots, (x'_{n'}, y'_{n'}))$ have $n = n'$ and they only differ in one index k^*.*

Document and sentence level neighborhoods Next, we assume that a text dataset D consists of documents $d_1, \ldots, d_{n_d}$, and each document d_i can be unambiguously split into sentences $s_{i,1}, \ldots, s_{i,n_{s,i}}$. For a second dataset D', we define d'_i, n'_d, $s'_{i,j}$ and $n'_{s,i}$ in the same way to denote documents, the number of documents, sentences and number of sentences in D', respectively.

Definition A.9. *Datasets D and D' are document-level zero-out neighbors, denoted $D \sim_d^{ZO} D'$, if $d_i = d'_i$ for all i except for at most one document i^*, and either $d_{i^*} = \perp$ or $d'_{i^*} = \perp$, where $\perp$ is a zero marker that makes any query return 0.*

Definition A.10. *Datasets D and D' are document-level add/remove neighbors, denoted $D \sim_d^{A/R} D'$, if they only differ by the addition or removal of at most one document.*

Definition A.11. *Datasets D and D' are document-level substitute neighbors, denoted $D \sim_d^{S} D'$, if $d_i = d'_i$ for all i except for at most one document i^*.*

Definition A.12. *Datasets D and D' are sentence-level zero-out neighbors, denoted $D \sim_s^{ZO} D'$, if $d_i = d'_i$ except for at most one document i^*, and $s_{i^*,j} = s'_{i^*,j}$ except for one sentence j^* in document i^*, and either $s_{i^*,j^*} = \perp$ or $s'_{i^*,j^*} = \perp$, where $\perp$ is a zero marker that makes any query return 0.*

Definition A.13. *Datasets D and D' are sentence-level add/remove neighbors, denoted $D \sim_s^{A/R} D'$, if $d_i = d'_i$ except for at most one document i^*, and the sequences $s_{i^*,1}, \ldots s_{i^*,n_{s,i}}$, $s'_{i^*,1}, \ldots s'_{i^*,n'_{s,i}}$ differ in the addition or removal of at most one sentence.*

Note that Definition A.13 does not allow creating a new document by the insertion of a sentence. It also does not allow removing a document by deleting a sentence. In case the only sentence in a document is deleted, the document remains as an empty document.

Definition A.14. *Datasets D and D' are sentence-level substitute neighbors, denoted $D \sim_s^{S} D'$, if $d_i = d'_i$ except for at most one document i^*, and $s_{i^*,j} = s'_{i^*,j}$ except for one sentence j^* in document i^*.*

B SEQUENCE-LEVEL CONVERSION FACTORS

This appendix collects the sequence-level conversion factors that instantiate Theorem 5.2 when the base and target neighborhood relations are induced by fixed-length token sequences of different lengths. We consider a fixed tokenized corpus and define different sequence-level neighborhood relations on this same underlying object by partitioning the token sequence into fixed-length chunks. Thus, changing the sequence length changes the neighborhood relation, not the dataset domain.

Let $\sim$ denote the base sequence-level neighborhood relation with sequence length L, and let $\sim_\star$ denote the target sequence-level neighborhood relation with sequence length $L_\star$. The following theorem gives valid values of $K = k(\sim \to \sim_\star)$ for Theorem 5.2. An entry $K = \infty$ means that the target-neighboring change cannot in general be represented by any finite number of base-neighboring changes.

Theorem B.1 (Sequence-level conversion factors)**.** *Consider conversions between sequence-level privacy with different sequence lengths, with length L for the base relation $\sim$ and length $L_\star$ for the target relation $\sim_\star$.*

1. *If $L_\star > L$, and the sequencing and tokenization are such that every $\sim_\star$-sequence overlaps with at most $r \in \mathbb{N}$ $\sim$-sequences. Then $K = k(\sim \to \sim_\star)$ can be chosen according to the following table, where rows denote the base adjacency relation $\sim$ and columns denote the target adjacency relation $\sim_\star$:*

$\sim$ / $\sim_\star$	*Substitute*	*Zero-out*	*Add/remove*
Substitute	$K = r$	$K = r$	$K = \infty$
Zero-out	$K = 2r$	$K = r$	$K = \infty$
Add/remove	$K = 2r$	$K = 2r$	$K = r$

If $L_\star$ is divisible by L and the sequencing and tokenization are aligned such that every $\sim$-sequence is contained in exactly one $\sim_\star$-sequence, then $r = L_\star/L$.

2. *(Aligned sequences) If $L_\star < L$, L is divisible by $L_\star$, and the sequencing and tokenization are aligned such that every $\sim_\star$-sequence is contained in exactly one $\sim$-sequence. Then $K = k(\sim \to \sim_\star)$ can be chosen according to the following table, where rows denote the base adjacency relation $\sim$ and columns denote the target adjacency relation $\sim_\star$:*

$\sim$ / $\sim_\star$	*Substitute*	*Zero-out*	*Add/remove*
Substitute	$K = 1$	$K = 1$	$K = \infty$
Zero-out	$K = 2$	$K = 2$	$K = \infty$
Add/remove	$K = 2$	$K = 2$	$K = 2$

3. *(Non-aligned sequences) If $L_\star < L$, then $K = k(\sim \to \sim_\star)$ can be chosen according to the following table, where rows denote the base adjacency relation $\sim$ and columns denote the target adjacency relation $\sim_\star$:*

$\sim$ / $\sim_\star$	*Substitute*	*Zero-out*	*Add/remove*
Substitute	$K = 2$	$K = 2$	$K = \infty$
Zero-out	$K = 4$	$K = 4$	$K = \infty$
Add/remove	$K = 4$	$K = 4$	$K = 4$

Proof. Proof of (1): going through the cells of the table:

- Diagonal of the table: in each case, a change according to $\sim_\star$ can be replicated with r changes according to $\sim$.
- $\sim$: substitute, $\sim_\star$: zero-out: both zeroing-out and replacing a zeroed-out sequence according to $\sim_\star$ can be replicated in r substitutions according to $\sim$.
- $\sim$: substitute, $\sim_\star$: add/remove: adding a sequence according to $\sim_\star$ cannot be replicated by any number of substitutions according to $\sim$.
- $\sim$: zero-out, $\sim_\star$: substitute: replicating a substitution according to $\sim_\star$ requires two operations according to $\sim$ on each of the r shorter sequences.
- $\sim$: zero-out, $\sim_\star$: add/remove: adding a sequence according to $\sim_\star$ cannot be replicated by any number of operations according to $\sim$.
- $\sim$: add/remove, $\sim_\star$: substitute: replicating a substitution according to $\sim_\star$ requires two operations according to $\sim$ on each of the r shorter sequences.
- $\sim$: add/remove, $\sim_\star$: zero-out: replicating a zero-out or replacement of a zeroed-out sequence according to $\sim_\star$ requires a remove and an add operation from $\sim$ for each of the r subsequences.

In the particular case that $L_\star$ is divisible by L and the sequencing and tokenization are aligned such that every $\sim$-sequence is contained in exactly one $\sim_\star$-sequence, clearly $r = L_\star/L$.

Proof of (2): in this case, a shorter sequence is changed according to $\sim_\star$, and a change according to $\sim$ in a longer sequence that contains all of the shorter sequence must replicate it. Going through each case:

- $\sim$: substitute, $\sim_\star$: substitute: a substitution of the shorter sequence can be replicated with a single substitution of the longer sequence.
- $\sim$: substitute, $\sim_\star$: zero-out: zeroing-out or undoing a zero-out in the shorter sequence can be replicated by substituting the longer sequence.
- $\sim$: substitute, $\sim_\star$: add-remove: adding a shorter sequence cannot be replicated by substituting any number of the original longer sequences.

- $\sim$: zero-out, $\sim_\star$: substitute: replicating a substitution of the shorter sequence requires first zeroing-out the longer sequence and then replacing the zeroed-out sequence to match the changed shorter sequence.
- $\sim$: zero-out, $\sim_\star$: zero-out: replicating a zero-out or undoing of a zero-out in the shorter sequence requires first zeroing-out the longer sequence and then replacing the zeroed-out sequence to match the changed shorter sequence.
- $\sim$: zero-out, $\sim_\star$: add/remove: adding a shorter sequence cannot be replicated by zeroing-out or undoing a zero-out on any number of the original longer sequences.
- $\sim$: add/remove, $\sim_\star$: substitute: replicating a substitution of the shorter sequence requires first removing the longer sequence and then adding it back to match the changed shorter sequence.
- $\sim$: add/remove, $\sim_\star$: zero-out: replicating a zero-out or undoing of a zero-out in the shorter sequence requires first removing the longer sequence and then adding it back to match the changed shorter sequence.
- $\sim$: add/remove, $\sim_\star$: add/remove: replicating a removal in the shorter sequence requires first removing the longer sequence and then adding it back with the shorter sequence removed. Replicating an addition of a shorter sequence only requires one addition of a sequence. Note that we allow $\sim$ to add sequences with lengths $\leq L$, which makes it possible to replicate the addition of an $L_\star$-length sequence.

Proof of (3): In this case, a shorter sequence that is changed according to $\sim_\star$ can overlap with at most two longer sequences according to $\sim$. Going through each case:

- $\sim$: substitute, $\sim_\star$: substitute: a substitution of a shorter sequence that overlaps two longer sequences requires two substitutions to replicate.
- $\sim$: substitute, $\sim_\star$: zero-out: zeroing-out or undoing a zero-out in a shorter sequence overlapping with two longer sequences can be replicated by two substitutions.
- $\sim$: substitute, $\sim_\star$: add/remove: adding a shorter sequence cannot be replicated by substituting any number of the original longer sequences.
- $\sim$: zero-out, $\sim_\star$: substitute: replicating a substitution of the shorter sequence requires first zeroing-out both overlapping longer sequences and then replacing both zeroed-out sequences to match the changed shorter sequence.
- $\sim$: zero-out, $\sim_\star$: zero-out: replicating a zero-out or undoing of a zero-out in the shorter sequence requires first zeroing-out both overlapping longer sequences and then replacing the zeroed-out sequences to match the changed shorter sequence.
- $\sim$: zero-out, $\sim_\star$: add/remove: adding a shorter sequence cannot be replicated by zeroing-out or undoing a zero-out on any number of the original longer sequences.
- $\sim$: add/remove, $\sim_\star$: substitute: replicating a substitution of the shorter sequence requires first removing both overlapping longer sequences and then adding them back to match the changed shorter sequence.
- $\sim$: add/remove, $\sim_\star$: zero-out: replicating a zero-out or undoing of a zero-out in the shorter sequence requires first removing both overlapping longer sequences and then adding them back to match the changed shorter sequence.
- $\sim$: add/remove, $\sim_\star$: add/remove: replicating a removal in the shorter sequence requires first removing both overlapping longer sequences and then adding them back with the shorter sequence removed. Replicating an addition of a shorter sequence only requires one addition of a sequence. Note that we allow $\sim$ to add sequences with lengths $\leq L$, which makes it possible to replicate the addition of an $L_\star$-length sequence.

□

C Missing Proofs for Section 4

Theorem 4.1. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be algorithms that are respectively $(f_t, \sim_t)$-DP. Let $\sim_\star$ be a target neighborhood relation. Under the assumptions in Equation (3), the composition of $\mathcal{M}_1, \ldots, \mathcal{M}_T$ is $(f_{\text{final}}, \sim_\star)$-DP with*

$$f_{\text{final}} := \operatorname*{lce}_{\substack{(a_1,\ldots,a_T)\in\mathbb{Z}^T_{\geq 0} \\ 0\leq a_t\leq A_t,\ \sum_{t=1}^T \mathbf{1}\{a_t>0\}\leq r,\ \sum_{t=1}^T a_t\leq B}} \bigotimes_{t=1}^T \Gamma_{a_t}(f_t). \tag{4}$$

Proof. Let $D_u \sim_\star D'_u$ be datasets that only differ in the unit u, and denote the composition of $\mathcal{M}_1, \ldots, \mathcal{M}_T$ by $\mathcal{M}^*$. We denote the output of $\mathcal{M}_t$ by θ_t. The composition is adaptive, so $\theta_t \sim \mathcal{M}(D_t, \theta_{<t})$, where $\theta_{<t} = (\theta_1, \ldots, \theta_{t-1})$. We cannot apply typical group privacy and composition results directly to bound $T(\mathcal{M}_t(D_u), \mathcal{M}_t(D'_u))$, since they are stated for the worst-case u, not an individual u. However, the proof of the f-DP group privacy theorem (Dong et al., 2022, Theorem 3) shows that it applies analogously to any u, which means

$$T(\mathcal{M}_t(D_u, \theta_{<t}), \mathcal{M}_t(D'_u, \theta_{<t})) \geq \Gamma_{a_t(u)}(f_t) \tag{7}$$

for all $\theta_{<t}$. Composition can be applied to any u in the same way (Dong et al., 2022, Lemma C.1), so

$$T(\mathcal{M}^*(D_u), \mathcal{M}^*(D'_u)) \geq \bigotimes_{t=1}^T T(\mathcal{M}_t(D_u, \theta_{<t}), \mathcal{M}_t(D'_u, \theta_{<t})) \geq \bigotimes_{t=1}^T \Gamma_{a_t(u)}(f_t). \tag{8}$$

With this, for the worst-case u and $D_u \sim_\star D'_u$, we have

$$\inf_{u, D_u\sim_\star D'_u} T(\mathcal{M}^*(D_u), \mathcal{M}^*(D'_u)) \geq \inf_{u, D_u\sim_\star D'_u} \bigotimes_{t=1}^T \Gamma_{a_t(u)}(f_t). \tag{9}$$

$\bigotimes_{t=1}^T \Gamma_{a_t(u)}(f_t)$ does not depend on D_u, D'_u beyond u, so they can be dropped from the infimum. u is constrained by Equation (3). Since all of the constraints concern $a_t(u)$ and the objective depends on u only through $a_t(u)$, optimizing over all u is equivalent to optimizing over integer vectors $(a_1, \ldots, a_T)$. There is a finite number of these vectors satisfying the constraints, so the infimum is in fact a minimum. The minimum of tradeoff functions is not necessarily a tradeoff function, but we can lower bound the minimum with the lower convex envelope, which results in a tradeoff function. Put together, these observations mean

$$\inf_{u, D_u\sim_\star D'_u} T(\mathcal{M}^*(D_u), \mathcal{M}^*(D'_u)) \geq \operatorname*{lce}_{\substack{(a_1,\ldots,a_T)\in\mathbb{Z}^T_{\geq 0} \\ 0\leq a_t\leq A_t,\ \sum_{t=1}^T \mathbf{1}\{a_t>0\}\leq r,\ \sum_{t=1}^T a_t\leq B}} \bigotimes_{t=1}^T \Gamma_{a_t}(f_t) \tag{10}$$

which proves the claim. □

Corollary 4.3. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be algorithms that are respectively $(\mu_t, \sim_t)$-GDP. Let $\sim_\star$ be a target neighborhood relation. Under the assumptions in Equation (3), the composition of $\mathcal{M}_1, \ldots, \mathcal{M}_T$ is $(\mu_{\text{final}}, \sim_\star)$-GDP with*

$$\mu^2_{\text{final}} := \max_{\substack{(a_1,\ldots,a_T)\in\mathbb{Z}^T_{\geq 0} \\ 0\leq a_t\leq A_t,\ \sum_{t=1}^T \mathbf{1}\{a_t>0\}\leq r,\ \sum_{t=1}^T a_t\leq B}} \sum_{t=1}^T a_t^2\mu_t^2. \tag{5}$$

Proof. μ-GDP is equivalent to G_μ-DP for function family G_μ that satisfies the following (Dong et al., 2022):

$$\mu \leq \mu' \Leftrightarrow G_\mu \geq G_{\mu'}, \tag{11}$$

$$\bigotimes_{t=1}^T G_{\mu_t} = G_{\sqrt{\sum_{t=1}^T \mu_t^2}}, \tag{12}$$

$$\Gamma_k(G_\mu) = G_{k\mu}. \tag{13}$$

With these properties, Theorem 4.1 implies the claim. □

Proposition 4.4. *When $\mu_t > 0$ for all $t \in \{1, \dots T\}$, any optimal solution $(a_1^\star, \dots, a_T^\star)$ to Equation (5) has all but at most one nonzero coordinate at their stage-wise caps. That is, for all but at most one stage t, $a_t^\star \in \{0, A_t\}$.*

Proof. We prove the claim by contradiction. Suppose that the optimal solution $(a_1^\star, \dots, a_T^\star)$ has two distinct coordinates i and j satisfying

$$a_i^\star \notin \{0, A_i\}, \qquad a_j^\star \notin \{0, A_j\}, \tag{14}$$

which is equivalent to

$$0 < a_i^\star < A_i, \qquad 0 < a_j^\star < A_j. \tag{15}$$

Then both of the operations

$$j \to i \colon (a_i, a_j) \mapsto (a_i + 1, a_j - 1) \tag{16}$$

and

$$i \to j \colon (a_i, a_j) \mapsto (a_i - 1, a_j + 1) \tag{17}$$

maintain the constraints. The respective changes in the objective are

$$\Delta_{j \to i}(a_1^\star, \dots, a_T^\star) = \mu_i^2(2a_i^\star + 1) - \mu_j^2(2a_j^\star - 1) \tag{18}$$

and

$$\Delta_{i \to j}(a_1^\star, \dots, a_T^\star) = \mu_j^2(2a_j^\star + 1) - \mu_i^2(2a_i^\star - 1). \tag{19}$$

Their sum is

$$\Delta_{j \to i}(a_1^\star, \dots, a_T^\star) + \Delta_{i \to j}(a_1^\star, \dots, a_T^\star) = 2(\mu_i^2 + \mu_j^2) > 0. \tag{20}$$

Hence, at least one of the two operations strictly increases the objective, so $(a_1^\star, \dots, a_T^\star)$ is not an optimal solution after all. □

Algorithm 1 Finding the worst-case μ_{final} under constraints.

Input: stage-wise privacy bound $\mu_1, \dots, \mu_T$, stage-wise contribution limits $A_1, \dots, A_T$, maximum number of contributed stages r, maximum number of contributions B
Output: μ_{final}
$\mu^2_{\text{current}} := 0$
for $0 \le s \le \min(r, T)$ **do**
  for subsets C of $\{1, \dots, T\}$ of size s **do**
    $B_{\text{rem}} := B - \sum_{j \in C} A_j$
    $\mu_C^2 := \sum_{j \in C} A_j^2 \mu_j^2$
    if $B_{\text{rem}} \ge 0$ **then**
      if $s < \min(r, T)$ **then**
        for $1 \le i \le T, i \notin C$ **do**
          $a := \min\{B_{\text{rem}}, A_i\}$
          $\mu^2 := \mu_C^2 + a^2 \mu_i^2$
          $\mu^2_{\text{current}} := \max\{\mu^2, \mu^2_{\text{current}}\}$
        end for
      else
        $\mu^2_{\text{current}} := \max\{\mu_C^2, \mu^2_{\text{current}}\}$
      end if
    end if
  end for
end for
Return μ_{current}

D MISSING PROOFS FOR SECTION 5

D.1 BLACK-BOX OPTIMALITY OF CONVERT-THEN-COMPOSE WITH FULL DATA OVERLAP

The following result records the sense in which separating conversion and composition is optimal in a black-box analysis.

Definition D.1 (Optimal conversion)**.** *Let $\mathcal{M}$ be an $(f_1, \sim_1)$-DP mechanism. We call the greatest trade-off function f_2 such that $\mathcal{M}$ is $(f_2, \sim_2)$-DP the optimal conversion of $\mathcal{M}$ from $(f_1, \sim_1)$ to $\sim_2$.*

Proposition D.2 (Conditional optimality of convert-then-compose)**.** *Let $\mathcal{M}_1$ be an $(f_1, \sim_1)$-DP mechanism and let $\mathcal{M}_2$ be an $(f_2, \sim_2)$-DP mechanism. Suppose f_1' is the optimal conversion of $\mathcal{M}_1$ to $\sim_\star$ and f_2' is the optimal conversion of $\mathcal{M}_2$ to $\sim_\star$. Then the adaptive sequential composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ is*

$$(f_1' \otimes f_2', \sim_\star)\text{-DP}. \tag{21}$$

Moreover, if the sequential composition bound is tight for the converted first-stage mechanism and the second-stage mechanism, then no strictly larger trade-off function can be certified from these two pieces of information alone.

Proof. After conversion, both mechanisms satisfy DP with respect to the same neighborhood relation, so the stated guarantees follow from adaptive sequential composition.

Let (P_1, Q_1) be a tight dominating pair (Zhu et al., 2022) for $\mathcal{M}_1$ under $\sim_1$, meaning that $T(P_1, Q_1) = f_1$, and let (P_2, Q_2) be defined likewise for $\mathcal{M}_2$. Let (P_1', Q_1') and (P_2', Q_2') be tight dominating pairs for $\mathcal{M}_1$ and $\mathcal{M}_2$ under $\sim_\star$, respectively. If both conversions from $\sim_1$ and $\sim_2$ to $\sim_\star$ are optimal, $T(P_1', Q_1') = f_1'$ and $T(P_2', Q_2') = f_2'$. We know that $(P_1' \times P_2', Q_1' \times Q_2')$ is a tight[2] dominating pair for the composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ under $\sim_\star$ (Zhu et al., 2022), and we know that (Dong et al., 2022)

$$T(P_1' \times P_2', Q_1' \times Q_2') = T(P_1', Q_1') \otimes T(P_2', Q_2') = f_1' \otimes f_2'. \tag{22}$$

Since $(P_1' \times P_2', Q_1' \times Q_2')$ is a tight dominating pair for the composition, we know that there cannot be an $f' > T(P_1' \times P_2', Q_1' \times Q_2') = f_1' \otimes f_2'$ such that the composition is f'-DP. □

Remark 1. The proposition is a black-box statement. It does not exclude sharper bounds for specific mechanisms when additional structure is known, for example details of the clipping rule, sampling scheme, or overlap pattern.

Remark 2. The optimality only holds when the composition is fully sequential, meaning that the worst-case privacy unit is in both pretraining and adaptation. If there is information on the overlapping and non-overlapping privacy units which implies that the worst-case unit cannot be in both stages, that can be exploited to obtain a smaller privacy bound than sequential composition, as in Theorem 4.1.

D.2 LEMMA FOR THEOREM 5.2

Lemma D.3. *Let $\sim_1$ and $\sim_2$ be neighborhood relations such that for all datasets D, D', $D \sim_1 D' \Rightarrow D \sim_2 D'$. Now, if an algorithm $\mathcal{M}$ is $(f, \sim_2)$-DP, it is $(f, \sim_1)$-DP.*

Proof. Let D, D' be any datasets with $D \sim_1 D'$. By the assumption, also $D \sim_2 D'$. By the assumption that $\mathcal{M}$ is $\sim_2$-DP, we have $T(\mathcal{M}(D), \mathcal{M}(D')) \geq f$, which concludes the proof. □

The same proof structure also applies to any other definition of DP.

D.3 LIMITATIONS OF IMPROVING CONVERSION FACTORS

The conversions in Theorem 5.2 and the factors in Appendix B are black-box bounds: they use only the stage-wise DP guarantee and the relationship between the base and target neighborhood relations. The next results show that some apparently conservative conversions cannot be improved in general.

First, when converting from smaller units to larger units, the f-DP and GDP results in Theorem 5.2 are tight in general.

[2] Zhu et al. (2022) do not include tightness in their composition theorem statement, but their proof shows that if the composed dominating pairs are tight, the composed pair is also tight.

Theorem D.4 (Tightness for smaller-to-larger conversion)**.** *Fix a positive integer $K \geq 2$, a universe of datasets $\mathcal{D}$ of at least size K, and a query ϕ such that $||\phi(x)|| \leq \Delta$ for any datapoint x, there exists a datapoint x with $\|\phi(x)\| = \Delta$, and a dataset can contain at least K copies of x. Also assume that for any datapoint x, there exists a datapoint x' such that $\phi(x) = -\phi(x')$. Let $\sim$ be example-level zero-out (or add/remove or substitute) neighborhood. Then there exists a neighborhood relation $\sim_\star$ such that every $\sim_\star$-neighboring pair differs by exactly K applications of $\sim$, and a mechanism $\mathcal{M}$ that is tightly $(\mu, \sim)$-GDP, tightly $(f, \sim)$-DP for some μ and f, tightly $(K\mu, \sim_\star)$-GDP, and tightly $(f_\star, \sim_\star)$-DP, where*

$$f_\star = \Gamma_K(f). \tag{23}$$

Proof. Let $\sim_\star$ relate datasets that differ by zeroing-out (or the addition or removal) of exactly K datapoints. Consider the Gaussian mechanism

$$\mathcal{M}(D) = \sum_{x \in D} \phi(x) + \mathcal{N}(0, \sigma^2), \tag{24}$$

with $\sigma = \Delta/\mu$. Its sensitivity under $\sim$ is Δ, so the tight GDP parameter is μ (Dong et al., 2022). Its sensitivity under $\sim_\star$ is $K\Delta$, attained by choosing K datapoints with query norm Δ in the same direction, so the tight GDP parameter is $K\mu$. Since GDP is a special case of f-DP and the group-privacy transformation for GDP satisfies $\Gamma_K(G_\mu) = G_{K\mu}$ (Dong et al., 2022), the corresponding f-DP statement follows.

The claim with substitute neighborhood is proven in the same way, except the sensitivities are twice as large. □

The reverse direction, from larger privacy units to smaller ones, also need not improve the certified privacy guarantee without additional assumptions on the mechanism. This is unintuitive because the smaller-unit notion is weaker: for example, item-level DP is weaker than user-level DP when a user may contribute many items. Nevertheless, if clipping is defined at the larger-unit level, a smaller-unit change can disrupt cancellation within a larger unit and thereby induce a larger change in the clipped contribution than a larger-unit change. The following result formalizes this phenomenon for zero-out sequence-level relations in the clipped adaptive batch linear queries model, which captures the standard assumptions used in DP-SGD privacy accounting (Chua et al., 2024b).

Algorithm 2 Clipped Adaptive Batch Linear Queries

Input: Data $(x_1, \ldots, x_n)$, batch selector B, adaptive queries A, noise scale σ, clip bound C
Output: Query values $(g_1, \ldots, g_T)$
for $1 \leq t \leq T$ **do**
  $S_t \sim B$
  $\psi_t(\cdot) \leftarrow A(g_1, \ldots, g_{t-1}, \cdot)$
  $\eta_t \sim \mathcal{N}(0, \sigma^2)$
  $g_t \leftarrow \sum_{i \in S_t} \mathrm{clip}_C(\psi_t(x_i)) + C\eta_t$
end for
return $(g_1, \ldots, g_T)$

Theorem D.5 (Larger-to-smaller zero-out conversion cannot in general have $K = 1$)**.** *For any sequence length $L_\star$ and $L = 2L_\star$, let $\sim_\star$ and $\sim$ be sequence-level zero-out neighborhoods for $L_\star$- and L-token sequences, respectively. Then there exists an instantiation of clipped adaptive batch linear queries (Algorithm 2), with clipping defined with respect to the L-token units, for which the worst-case change in the clipped contribution under $\sim_\star$ is strictly larger than under $\sim$. Consequently, one cannot in general obtain the smaller-unit guarantee by setting $K = 1$ in the conversion rules of Theorem 5.2.*

Proof. It suffices to construct a query for which the clipped contribution changes by more under a short-sequence zero-out than under a long-sequence zero-out. Let one L-token sequence consist of two consecutive $L_\star$-token halves. Consider a single affected sequence x, and choose a query ψ such that the first half contributes $2C$ and the second half contributes $-C$. Then the full-sequence contribution is

$$\psi(x) = 2C - C = C, \tag{25}$$

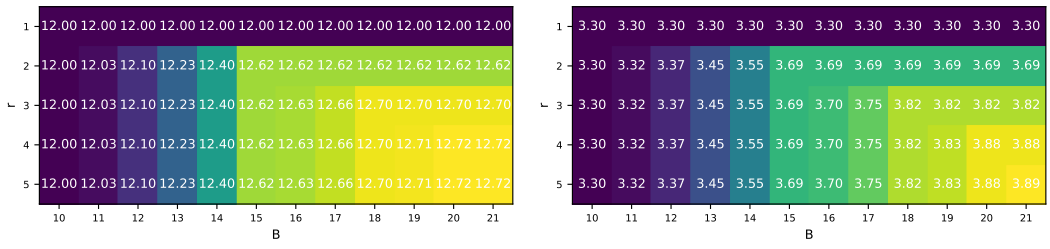
so clipping at threshold C does not reduce it.

Now compare the two neighborhood relations. Under the long-sequence zero-out relation $\sim$, the whole sequence is replaced by zero, so the clipped contribution changes from C to 0, for a change of magnitude C. Under the short-sequence zero-out relation $\sim_\star$, only the first half is zeroed out, so the contribution changes from C to $-C$, for a change of magnitude $2C$. Thus the worst-case change in the clipped contribution can be strictly larger under $\sim_\star$ than under $\sim$.

Therefore the conversion from the larger unit to the smaller unit cannot in general be certified with $K = 1$ using only the black-box argument of Theorem 5.2. A full construction is obtained by taking a single query and a single affected sequence as above. □

Intuition. The phenomenon is analogous to the gap between user-level and item-level privacy when clipping is performed per user. Although item-level privacy is a weaker notion than user-level privacy, deleting a single item can perturb the clipped per-user contribution more than deleting the whole user if the user's items partially cancel before clipping. Here, the larger L-token sequence plays the role of the user, the shorter $L_\star$-token subsequence plays the role of the item, and clipping is performed at the larger-unit level.

E Extra Results



(a) $\mu_{1:T} = (1.2, 0.78, 0.49, 0.33, 0.17)$ (b) $\mu_{1:T} = (0.33, 0.33, 0.33, 0.33, 0.33)$

Figure 3: Extra settings of $\mu_{1:T}$ for the optimization problem from Section 6. In (a), we put the μ-values in decreasing order, reflecting a case where pretraining is allocated the largest privacy budget and subsequent stages have decreasing budgets. In this case, the cost of pretraining dominates even more heavily than in any setting from Figure 2. In (b), we have a constant μ budget for all stages. The result is similar to panel (a) of Figure 2 which was based on privacy budgets typically used for pretraining and finetuning in the literature.