How Private are DP-SGD Implementations?

Lynn Chua [1] **Badih Ghazi** [1] **Pritish Kamath** [1] **Ravi Kumar** [1] **Pasin Manurangsi** [1] **Amer Sinha** [1]
Chiyuan Zhang [1]

Abstract

We demonstrate a substantial gap between the privacy guarantees of the **A**daptive **B**atch **L**inear **Q**ueries (ABLQ) mechanism under different types of batch sampling: (i) Shuffling, and (ii) Poisson subsampling; the typical analysis of **D**ifferentially **P**rivate **S**tochastic **G**radient **D**escent (DP-SGD) follows by interpreting it as a post-processing of ABLQ. While shuffling-based DP-SGD is more commonly used in practical implementations, it has not been amenable to easy privacy analysis, either analytically or even numerically. On the other hand, Poisson subsampling-based DP-SGD is challenging to scalably implement, but has a well-understood privacy analysis, with multiple open-source numerically tight privacy accountants available. This has led to a common practice of using shuffling-based DP-SGD in practice, but using the privacy analysis for the corresponding Poisson subsampling version. Our result shows that there can be a substantial gap between the privacy analysis when using the two types of batch sampling, and thus advises caution in reporting privacy parameters for DP-SGD.

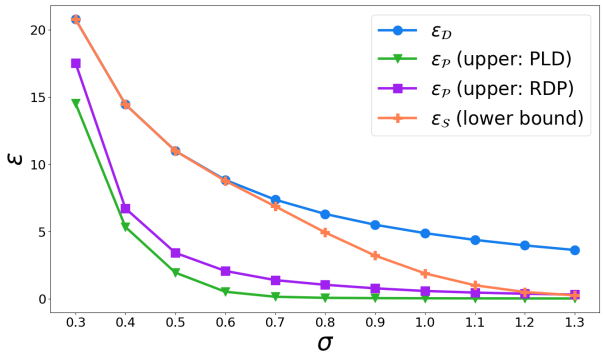

Figure 1. Privacy parameter ε for different noise parameters σ, for fixed $\delta = 10^{-6}$ and number of steps $T = 10,000$. $\varepsilon_{\mathcal{D}}$: for deterministic batching, $\varepsilon_{\mathcal{P}}$: upper bounds when using Poisson subsampling (computed using different accountants), and $\varepsilon_{\mathcal{S}}$: a lower bound when using shuffling. We observe that shuffling does not provide much amplification for small values of σ, incurring significantly higher privacy cost compared to Poisson subsampling.

1. Introduction

Using noisy gradients in first-order methods such as stochastic gradient descent (SGD) has become a prominent approach for adding differential privacy (DP) to the training of differentiable models such as neural networks. This approach, introduced by Abadi et al. (2016), has come to be known as Differentially Private Stochastic Gradient Descent, and we use the term DP-SGD to refer to any such first-order method. DP-SGD is currently the canonical algorithm for training deep neural networks with privacy guarantees, and there currently exist multiple open source implementations, such as in Tensorflow Privacy (a), Pytorch Opacus (Yousefpour et al., 2021), and JAX Privacy (Balle et al., 2022). The algorithm has been applied widely in various machine learning domains, such as training of image classification (Tramer & Boneh, 2021; Papernot et al., 2021; Klause et al., 2022; De et al., 2022; Bu et al., 2022), generative models with GAN (Torkzadehmahani et al., 2019; Chen et al., 2020), diffusion models (Dockhorn et al., 2022), language models (Li et al., 2022; Yu et al., 2022; Anil et al., 2022; He et al., 2023), medical imaging (Ziller et al., 2021), as well as private spatial querying (Zeighami et al., 2022), ad modeling (Denison et al., 2023), and recommendation (Fang et al., 2022).

DP-SGD operates by processing the training data in mini-batches, and at each step, performs a first-order gradient update, using a noisy estimate of the average gradient for each mini-batch. In particular, the gradient g for each record is first clipped to have a pre-determined bounded ℓ_2-norm, by setting $[g]_C := g \cdot \min\{1, C/\|g\|_2\}$, and then adding Gaussian noise of standard deviation σC to all coordinates of the sum of gradients in the mini-batch.[1] The privacy

[1]Google Research. Correspondence to: Pritish Kamath <pritish@alum.mit.edu>, Pasin Manurangsi <pasin@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]While some distributed training setup uses the *sum* gradient directly, it is common to rescale the *sum* gradient by the batch size

Algorithm 1 $\mathrm{ABLQ}_{\mathcal{B}}$: Adaptive Batch Linear Queries

Parameters: Batch sampler $\mathcal{B}$ using (expected) batch size b and number of batches T, noise parameter σ, and an (adaptive) query method $\mathcal{A} : (\mathbb{R}^d)^* \times \mathcal{X} \to \mathbb{B}^d$.
Input: Dataset $\boldsymbol{x} = (x_1, \ldots, x_n)$.
Output: Query estimates $g_1, \ldots, g_T \in \mathbb{R}^d$

$(S_1, \ldots, S_T) \leftarrow \mathcal{B}_{b,T}(n)$
for $t = 1, \ldots, T$ **do**
  $\psi_t(\cdot) := \mathcal{A}(g_1, \ldots, g_{t-1}; \cdot)$
  $g_t \leftarrow \sum_{i \in S_t} \psi_t(x_i) + e_t$ for $e_t \sim \mathcal{N}(0, \sigma^2 I_d)$
return $(g_1, \ldots, g_T)$

Algorithm 2 $\mathcal{D}_{b,T}$: Deterministic Batch Sampler

Parameters: Batch size b, number of batches T.
Input: Number of datapoints $n = b \cdot T$.
Output: Seq. of disjoint batches $S_1, \ldots, S_T \subseteq [n]$.
for $t = 0, \ldots, T-1$ **do**
  $S_{t+1} \leftarrow \{tb + 1, \ldots, tb + b\}$
return $S_1, \ldots, S_T$

Algorithm 3 $\mathcal{S}_{b,T}$: Shuffle Batch Sampler

Parameters: Batch size b, number of batches T.
Input: Number of datapoints $n = b \cdot T$.
Output: Seq. of disjoint batches $S_1, \ldots, S_T \subseteq [n]$.
Sample a random permutation π over $[n]$.
for $t = 0, \ldots, T-1$ **do**
  $S_{t+1} \leftarrow \{\pi(tb + 1), \ldots, \pi(tb + b)\}$
return $S_1, \ldots, S_T$

Algorithm 4 $\mathcal{P}_{b,T}$: Poisson Batch Sampler

Parameters: Expected batch size b, number of batches T.
Input: Number of datapoints n.
Output: Seq. of batches $S_1, \ldots, S_T \subseteq [n]$.
for $t = 1, \ldots, T$ **do**
  $S_t \leftarrow \emptyset$
  for $i = 1, \ldots, n$ **do**

$$S_t \leftarrow \begin{cases} S_t \cup \{i\} & \text{with probability } b/n \\ S_t & \text{with probability } 1 - b/n \end{cases}$$

return $S_1, \ldots, S_T$

guaranteed by the mechanism depends on the following: the choice of σ, the size of dataset, the size of mini-batches, the number of steps of gradient update performed, and finally *the process used to generate the batches*. Almost all deep learning systems generate fixed-sized batches of data by going over the dataset sequentially. When feasible, a global shuffling of all the examples in the dataset is performed for each training epoch by making a single pass over the dataset. On the other hand, the process analyzed by Abadi et al. (2016) constructs each batch by including each record with a certain probability, chosen i.i.d. However, this leads to variable-sized mini-batches, which is technically challenging to handle in practice. As a result, there is generally a mismatch between the actual training pipeline and the privacy accounting in many applications of DP-SGD, with the implicit assumption that this subtle difference is negligible and excusable. However, in this paper, we show that this is not true—in a typical setting, as shown in Figure 1, the privacy loss from the correct accounting is significantly larger than expected.

Adaptive Batch Linear Queries and Batch Samplers. Formally, the privacy analysis of DP-SGD, especially in the case of non-convex differentiable models, is performed by viewing it as a post-processing of a mechanism performing *adaptive batch linear queries* ($\mathrm{ABLQ}_{\mathcal{B}}$) as defined in Algorithm 1 (we use subscript $\mathcal{B}$ to emphasize the role of the batch sampler), where the linear query $\psi_t(x_i)$ corresponds to the clipped gradient corresponding to record x_i. For any x, we require that $\psi_t(x) \in \mathbb{B}^d := \{w \in \mathbb{R}^d : \|w\|_2 \leq 1\}$. Note that without loss of generality, we can treat the norm bound C to be 1 by defining $\psi_t(x) := [g]_C/C$ for the corresponding gradient g, and rescaling g_t by C to get back the noisy gradient for DP-SGD.

As mentioned above, a canonical way to generate the mini-batches is to go through the dataset in a fixed deterministic order, and divide the data into mini-batches of a fixed size (Algorithm 2, denoted as $\mathcal{D}$). Another commonly used option is to first randomly permute the entire dataset before dividing it into mini-batches of a fixed size (Algorithm 3, denoted as $\mathcal{S}$); this option provides *amplification by shuffling*, namely, that the privacy guarantees are better compared to fixed deterministic ordering. However, obtaining such amplification bounds is non-trivial, and while some bounds have been recently established for such privacy amplification (Erlingsson et al., 2019; Feldman et al., 2021; 2023), they tend to be loose in our setting and only kick in when the basic mechanism is already sufficiently private.

Instead, the approach followed by (Abadi et al., 2016), and henceforth used commonly in reporting privacy parameters for DP-SGD, is to assume that each batch is sampled i.i.d. by including each record with a certain probability, referred to as *Poisson subsampling* (Algorithm 4, denoted as $\mathcal{P}$). The advantage of this approach is that the privacy analysis of such sampling is easier to carry out since the $\mathrm{ABLQ}_{\mathcal{P}}$ mechanism can be viewed as a composition of T independent sub-mechanisms. This enables privacy accounting methods

to obtain the *average* gradient before applying the optimization step. Since this scaling factor can be assimilated in the learning rate, we focus on the *sum* gradient for simplicity in this paper. Note that, in case of DP-SGD using Poisson subsampling, the scaling is done by the *expected* batch size, and not the realized batch size.

such as Rényi DP (Mironov, 2017) as well as numerically tight accounting methods using privacy loss distributions (elaborated on later in Section 3).

This has been the case, even when the algorithm being implemented in fact uses some form of shuffling-based batch sampler. To quote Abadi et al. (2016) (with our emphasis),

> "In practice, for efficiency, the construction of batches and lots is done by randomly permuting the examples and then *partitioning* them into groups of the appropriate sizes. *For ease of analysis*, however, we assume that *each lot is formed by independently picking each example with probability* $q = L/N$, where N is the size of the input dataset."

Most implementations of DP-SGD mentioned earlier also use some form of shuffling, with a rare exception of PyTorch Opacus (Yousefpour et al., 2021) that has the option of Poisson subsampling to be consistent with the privacy analysis. But this approach does not scale to large datasets as random access for datasets that do not fit in memory is generally inefficient. Moreover, variable batch sizes are inconvenient to handle in deep learning systems[2]. Tensorflow Privacy (b) provides the `compute_dp_sgd_privacy_statement` method for computing the privacy parameters and reminds users about the implicit assumption of Poisson subsampling. Ponomareva et al. (2023) note in their survey that "It is common, though inaccurate, to train without Poisson subsampling, but to report the stronger DP bounds as if amplification was used."

As DP-SGD is being deployed in more applications with such discrepancy, it has become crucial to understand exactly how the privacy guarantee depends on the precise choice of the batch sampler, especially when one cares about specific (ε, δ) privacy parameters (and not asymptotic bounds). This leads us to our main motivating question:

> *How do the privacy guarantees of* $\mathrm{ABLQ}_{\mathcal{B}}$ *compare when using different batch samplers* $\mathcal{B}$*?*

1.1. Our Contributions

We study the privacy guarantees of $\mathrm{ABLQ}_{\mathcal{B}}$ for different choices of batch samplers $\mathcal{B}$. While we defer the formal definition of (ε, δ)-DP to Section 2, let $\delta_{\mathcal{B}}(\varepsilon)$ denote the *privacy loss curve* of $\mathrm{ABLQ}_{\mathcal{B}}$ for any $\mathcal{B} \in \{\mathcal{D}, \mathcal{P}, \mathcal{S}\}$, for a fixed choice of σ and T. Namely, for all $\varepsilon > 0$, let $\delta_{\mathcal{B}}(\varepsilon)$ be the smallest $\delta \geq 0$ such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP for any underlying adaptive query method $\mathcal{A}$. Let $\varepsilon_{\mathcal{B}}(\delta)$ be defined analogously.

$\mathcal{D}$ vs $\mathcal{S}$. We observe that $\mathrm{ABLQ}_{\mathcal{S}}$ always satisfies stronger privacy guarantees than $\mathrm{ABLQ}_{\mathcal{D}}$, i.e., $\delta_{\mathcal{S}}(\varepsilon) \leq \delta_{\mathcal{D}}(\varepsilon)$ for all $\varepsilon \geq 0$.

$\mathcal{D}$ vs $\mathcal{P}$. We show that the privacy guarantee of $\mathrm{ABLQ}_{\mathcal{D}}$ and $\mathrm{ABLQ}_{\mathcal{P}}$ are incomparable. Namely, for all values of T (number of steps) and σ (noise parameter), it holds (i) for small enough $\varepsilon > 0$ that $\delta_{\mathcal{P}}(\varepsilon) < \delta_{\mathcal{D}}(\varepsilon)$, but perhaps more interestingly, (ii) for sufficiently large ε it holds that $\delta_{\mathcal{P}}(\varepsilon) \gg \delta_{\mathcal{D}}(\varepsilon)$. We also demonstrate this separation in a specific numerical setting of parameters.

$\mathcal{S}$ vs $\mathcal{P}$. By combining the above it follows that for sufficiently large ε, it holds that $\delta_{\mathcal{S}}(\varepsilon) < \delta_{\mathcal{P}}(\varepsilon)$. If $\delta_{\mathcal{S}}(\varepsilon) < \delta_{\mathcal{P}}(\varepsilon)$ were to hold for all $\varepsilon > 0$, then reporting privacy parameters for $\mathrm{ABLQ}_{\mathcal{P}}$ would provide correct, even if pessimistic, privacy guarantees for $\mathrm{ABLQ}_{\mathcal{S}}$. However, we demonstrate multiple concrete settings of parameters, for which $\delta_{\mathcal{P}}(\varepsilon) \ll \delta_{\mathcal{S}}(\varepsilon)$, or alternately, $\varepsilon_{\mathcal{P}}(\delta) \ll \varepsilon_{\mathcal{S}}(\delta)$.

For example, in Figure 1, we fix $\delta = 10^{-6}$ and the number of steps $T = 10,000$,[3] and compare the value of $\varepsilon_{\mathcal{B}}(\delta)$ for various values of σ. For $\sigma = 0.5$, we find $\varepsilon_{\mathcal{P}}(\delta) < 1.96$ (PLD) and $\varepsilon_{\mathcal{P}}(\delta) < 3.43$ (RDP), but $\varepsilon_{\mathcal{S}}(\delta) > 10.994$ and $\varepsilon_{\mathcal{D}}(\delta) \approx 10.997$. For $\sigma = 1.3$, we find $\varepsilon_{\mathcal{P}}(\delta) < 0.031$ (PLD), whereas, $\varepsilon_{\mathcal{S}}(\delta) > 0.26$.

This suggests that reporting privacy guarantees using the Poisson batch sampler can significantly underestimate the privacy loss when the implementation in fact uses the shuffle batch sampler.

Our main takeaway is that batch sampling plays a crucial in determining the privacy guarantees of $\mathrm{ABLQ}_{\mathcal{B}}$, and hence caution must be exercised in reporting privacy parameters for mechanisms such as DP-SGD.

1.2. Technical Overview

Our techniques relies on the notion of *dominating pairs* as defined by Zhu et al. (2022) (see Definition 2.3), which if *tightly dominating* captures the privacy loss curve $\delta_{\mathcal{B}}(\varepsilon)$.

$\mathcal{D}$ vs $\mathcal{S}$. We observe that applying a mechanism on a random permutation of the input dataset does not degrade its privacy guarantees. While standard, we include a proof for completeness.

$\mathcal{D}$ vs $\mathcal{P}$. In order to show that $\delta_{\mathcal{P}}(\varepsilon) < \delta_{\mathcal{D}}(\varepsilon)$ for small enough $\varepsilon > 0$, we first show that $\delta_{\mathcal{P}}(0) < \delta_{\mathcal{D}}(0)$ by showing that the total variation distance between the tightly dominating pair for $\mathrm{ABLQ}_{\mathcal{D}}$ is larger than that in the case of

[2] For example, when the input shape changes, `jax.jit` will trigger recompilation, and `tf.function` will retrace the graph. Google TPUs require all operations to have fixed (input and output) shapes. Moreover, in various form of data parallelism, the batch size needs to be divisible by the number of accelerators.

[3] Recall that since we are analyzing a "single epoch", the subsampling probability of Poisson subsampling is $b/n = 1/T$.

$\mathrm{ABLQ}_{\mathcal{P}}$. And thus, by the continuity of hockey stick divergence in ε, we obtain the same for small enough ε.

In order to show that $\delta_{\mathcal{P}}(\varepsilon) > \delta_{\mathcal{D}}(\varepsilon)$ for large enough $\varepsilon > 0$, we demonstrate an explicit set E (a halfspace), which realizes a lower bound on the hockey stick divergence between the tightly dominating pair of $\mathrm{ABLQ}_{\mathcal{P}}$, and show that this decays slower than the hockey stick divergence between the tightly dominating pair for $\mathrm{ABLQ}_{\mathcal{D}}$.

We demonstrate the separation in a specific numerical setting of parameters using the `dp_accounting` library (Google's DP Library., 2020) to provide lower bounds on the hockey stick divergence between the tightly dominating pair for $\mathrm{ABLQ}_{\mathcal{P}}$.

$\mathcal{S}$ vs $\mathcal{P}$. One challenge in understanding the privacy guarantee of $\mathrm{ABLQ}_{\mathcal{S}}$ is the lack of a clear dominating pair for the mechanism. Nevertheless, we consider a specific instance of the query method $\mathcal{A}$, and a specific adjacent pair $\boldsymbol{x} \sim \boldsymbol{x}'$. The key insight we use in constructing this $\mathcal{A}$ and $\boldsymbol{x}, \boldsymbol{x}'$ is that in the case of $\mathrm{ABLQ}_{\mathcal{S}}$, since the batches are of a fixed size, the responses to queries on batches containing only the non-differing records in $\boldsymbol{x}$ and $\boldsymbol{x}'$ can *leak* information about the location of the differing record in the shuffled order. This limitation is not faced by $\mathrm{ABLQ}_{\mathcal{P}}$, since each record is independently sampled in each batch. We show a lower bound on the hockey stick divergence between the output distribution of the mechanism on these adjacent inputs, by constructing an explicit set E, thereby obtaining a lower bound on $\delta_{\mathcal{S}}(\varepsilon)$.

In order to show that this can be significantly larger than the privacy guarantee of $\mathrm{ABLQ}_{\mathcal{P}}$, we again use the `dp_accounting` library, this time to provide upper bounds on the hockey stick divergence between the dominating pair of $\mathrm{ABLQ}_{\mathcal{P}}$.

We provide the IPython notebook[4] that was used for all the numerical demonstrations; the notebook can be executed using a free CPU runtime on Google Colab.

Related work. The phenomenon of *non-differing* records leaking information about whether the differing record is in a batch or not can also exist in sampling-based batch samplers, such as, sampling independent batches of fixed-size, as studied in a concurrent work by Lebeda et al. (2024). However, we focus on the shuffle-based batch sampler, as these are most common in practical implementations.

2. Differential Privacy

We consider mechanisms that map input datasets to distributions over an output space, namely $\mathcal{M} : \mathcal{X}^* \to \Delta_{\mathcal{O}}$. That is, on input *dataset* $\boldsymbol{x} = (x_1, \ldots, x_n)$ where each *record* $x_i \in \mathcal{X}$, $\mathcal{M}(\boldsymbol{x})$ is a probability distribution over the output space $\mathcal{O}$; we abuse notation to also use $\mathcal{M}(\boldsymbol{x})$ to denote the underlying random variable. Two datasets $\boldsymbol{x}$ and $\boldsymbol{x}'$ are said to be *adjacent*, denoted $\boldsymbol{x} \sim \boldsymbol{x}'$, if, loosely speaking, they "differ in one record". This can be formalized in multiple ways, which we elaborate on in Section 2.2, but for any notion of adjacency, Differential Privacy (DP) can be defined as follows.

Definition 2.1 (DP)**.** For $\varepsilon, \delta \geq 0$, a mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if for all "adjacent" datasets $\boldsymbol{x} \sim \boldsymbol{x}'$, and for any (measurable) event E it holds that

$$\Pr[\mathcal{M}(\boldsymbol{x}) \in E] \;\leq\; e^{\varepsilon} \Pr[\mathcal{M}(\boldsymbol{x}') \in E] + \delta.$$

For any mechanism $\mathcal{M}$, let $\delta_{\mathcal{M}} : \mathbb{R}_{\geq 0} \to [0, 1]$ be its *privacy loss curve*, namely $\delta_{\mathcal{M}}(\varepsilon)$ is the smallest δ for which $\mathcal{M}$ satisfies (ε, δ)-DP; $\varepsilon_{\mathcal{M}} : [0, 1] \to \mathbb{R}_{\geq 0}$ can be defined analogously.

2.1. Adaptive Batch Linear Queries Mechanism

We primarily study the *adaptive batch linear queries* mechanism $\mathrm{ABLQ}_{\mathcal{B}}$ using a *batch sampler* $\mathcal{B}$ and an *adaptive query method* $\mathcal{A}$, as defined in Algorithm 1. The batch sampler $\mathcal{B}$ can be instantiated with any algorithm that produces a (randomized) sequence $S_1, \ldots, S_T \subseteq [n]$ of batches, where n is the number of examples. $\mathrm{ABLQ}_{\mathcal{B}}$ produces a sequence $(g_1, \ldots, g_T)$ of responses where each $g_i \in \mathbb{R}^d$. The response g_t is produced recursively using the adaptive query method $\mathcal{A}$ that given $g_1, \ldots, g_{t-1}$, constructs a new query $\psi_t : \mathcal{X} \to \mathbb{B}^d$ (for $\mathbb{B}^d := \{v \in \mathbb{R}^d : \|v\|_2 \leq 1\}$), and we estimate the sum of $\psi_t(x)$ over the batch S_t with added zero-mean Gaussian noise of scale σ to all coordinates. As explained in Section 1, DP-SGD falls under this abstraction by considering an adaptive query method that is specified by a differentiable loss function $f : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}$, and starts with an initial $w_0 \in \mathbb{R}^d$, and defines the query $\mathcal{A}(g_1, \ldots, g_{t-1}; x)$ as the clipped gradient $[\nabla_w f_{w_{t-1}}(x)]_1$ where w_t is the tth model iterate recursively obtained by performing gradient descent, e.g., $w_t \leftarrow w_{t-1} - \eta_t g_t$ (or any other first-order optimization step).

We consider the Deterministic $\mathcal{D}$ (Algorithm 2), Poisson $\mathcal{P}$ (Algorithm 4) and Shuffle $\mathcal{S}$ (Algorithm 3) batch samplers. As used in Section 1, we will continue to use $\delta_{\mathcal{B}}(\varepsilon)$ as a shorthand for denoting the *privacy loss curve* of $\mathrm{ABLQ}_{\mathcal{B}}$ for any $\mathcal{B} \in \{\mathcal{D}, \mathcal{P}, \mathcal{S}\}$. Namely, for all $\varepsilon > 0$, let $\delta_{\mathcal{B}}(\varepsilon)$ be the smallest $\delta \geq 0$ such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP for *all* choices of the underlying adaptive query method $\mathcal{A}$. And $\varepsilon_{\mathcal{B}}(\delta)$ is defined analogously.

2.2. Adjacency Notions

As alluded to earlier, the notion of adjacency is crucial to Definition 2.1. Commonly used adjacency notions are:

[4] https://colab.research.google.com/drive/1zI2H8YEXbQyD6gZVVskFwcOiM5YMvqRe?usp=sharing

Add-Remove adjacency. Datasets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}^*$ are said to be *add-remove* adjacent if there exists an i such that $\boldsymbol{x}' = \boldsymbol{x}_{-i}$ or vice-versa (where $\boldsymbol{x}_{-i}$ represents the dataset obtained by removing the ith record in $\boldsymbol{x}$).

Substitution adjacency. Datasets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}^*$ are said to be *substitution* adjacent if there exists an i such that $\boldsymbol{x}'_{-i} = \boldsymbol{x}_{-i}$ and $x'_i \neq x_i$.

The privacy analysis of DP-SGD is typically done for the Poisson batch sampler $\mathcal{P}$ (Abadi et al., 2016; Mironov, 2017), with respect to the *add-remove* adjacency. However, it is impossible to analyze the privacy of $\mathrm{ABLQ}_{\mathcal{D}}$ or $\mathrm{ABLQ}_{\mathcal{S}}$ with respect to the *add-remove* adjacency because the batch samplers $\mathcal{D}$ and $\mathcal{S}$ require that the number of records n equals $b \cdot T$. On the other hand, using the *substitution* adjacency for $\mathcal{D}$ and $\mathcal{S}$ leads to an unfair comparison to $\mathrm{ABLQ}_{\mathcal{P}}$ whose analysis is with respect to the *add-remove* adjacency. Thus, to make a fair comparison, we consider the following adjacency (proposed by Kairouz et al. (2021)).

Zero-out adjacency. We augment the input space to be $\mathcal{X}_\perp := \mathcal{X} \cup \{\perp\}$ and extend any adaptive query method $\mathcal{A}$ as $\mathcal{A}(g_1, \ldots, g_t; \perp) = \mathbf{0}$ for all $g_1, \ldots, g_t \in \mathbb{R}^d$. Datasets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_\perp^n$ are said to be *zero-out* adjacent if there exists i such that $\boldsymbol{x}_{-i} = \boldsymbol{x}'_{-i}$, and exactly one of $\{x_i, x'_i\}$ is in $\mathcal{X}$ and the other is $\perp$. Whenever we need to specifically emphasize that $x_i \in \mathcal{X}$ and $x'_i = \perp$, we will denote it as $\boldsymbol{x} \to_z \boldsymbol{x}'$. In this notation, $\boldsymbol{x} \sim \boldsymbol{x}'$ if either $\boldsymbol{x} \to_z \boldsymbol{x}'$ or $\boldsymbol{x}' \to_z \boldsymbol{x}$.

The privacy analysis of $\mathrm{ABLQ}_{\mathcal{P}}$ with respect to zero-out adjacency is the same as that with respect to the add-remove adjacency; it is essentially replacing a record by a "ghost" record that makes the query method always return $\mathbf{0}$. In the rest of this paper, we only consider this zero-out adjacency.

We note that our separations where $\varepsilon_{\mathcal{P}}(\delta) \ll \varepsilon_{\mathcal{S}}(\varepsilon)$, such as in Figure 1, also hold under the substitution adjacency, by using "group privacy", namely if a mechanism satisfies (ε, δ)-DP with respect to zero-out adjacency, then it satisfies $(2\varepsilon, \delta \cdot (1 + e^{\varepsilon}))$-DP (see, e.g., (Vadhan, 2017)).

2.3. Hockey Stick Divergence

We interchangeably use the same notation (e.g., letters such as P) to denote both a probability distribution and its corresponding density function. For $\mu \in \mathbb{R}^D$ and positive semi-definite $\Sigma \in \mathbb{R}^{D \times D}$, we use $\mathcal{N}(\mu, \Sigma)$ to denote the Gaussian distribution with mean μ and covariance Σ. For probability densities P and Q, we use $\alpha P + \beta Q$ to denote the weighted sum of the corresponding densities. $P \otimes Q$ denotes the product distribution sampled as (u, v) for $u \sim P$, $v \sim Q$, and, $P^{\otimes T}$ denotes the T-fold product distribution $P \otimes \cdots \otimes P$.

Definition 2.2. For all $\varepsilon \in \mathbb{R}$, the *$e^{\varepsilon}$-hockey stick divergence* between P and Q is $D_{e^{\varepsilon}}(P \| Q) := \sup_E \{P(E) - e^{\varepsilon} Q(E)\}$.

It is immediate to see that $\mathcal{M}$ satisfies (ε, δ)-DP iff for all adjacent $\boldsymbol{x} \sim \boldsymbol{x}'$, it holds that $D_{e^{\varepsilon}}(\mathcal{M}(\boldsymbol{x}) \| \mathcal{M}(\boldsymbol{x}')) \leq \delta$.

Definition 2.3 (Dominating Pair (Zhu et al., 2022))**.** The pair (P, Q) *dominates* the pair (A, B) if $D_{e^{\varepsilon}}(P \| Q) \geq D_{e^{\varepsilon}}(A \| B)$ holds for all $\varepsilon \in \mathbb{R}$. We say that (P, Q) *dominates* a mechanism $\mathcal{M}$ if (P, Q) dominates $(\mathcal{M}(\boldsymbol{x}), \mathcal{M}(\boldsymbol{x}'))$ for all adjacent $\boldsymbol{x} \to_z \boldsymbol{x}'$.

If (P, Q) dominates $\mathcal{M}$, then for all $\varepsilon \geq 0$, $\delta_{\mathcal{M}}(\varepsilon) \leq \max\{D_{e^{\varepsilon}}(P \| Q), D_{e^{\varepsilon}}(Q \| P)\}$. We say that (P, Q) *tightly dominates* a mechanism $\mathcal{M}$ if (P, Q) dominates $\mathcal{M}$ and there exist adjacent datasets $\boldsymbol{x} \to_z \boldsymbol{x}'$ such that $D_{e^{\varepsilon}}(P \| Q) = D_{e^{\varepsilon}}(\mathcal{M}(\boldsymbol{x}) \| \mathcal{M}(\boldsymbol{x}'))$ holds for all $\varepsilon \in R$ (note that this includes $\varepsilon < 0$); in this case, $\delta_{\mathcal{M}}(\varepsilon) = \max\{D_{e^{\varepsilon}}(P \| Q), D_{e^{\varepsilon}}(Q \| P)\}$. Thus, tightly dominating pairs completely characterize the privacy loss of a mechanism (although they are not guaranteed to exist for all mechanisms).[5] Dominating pairs behave nicely under mechanism compositions. Namely, if (P_1, Q_1) dominates $\mathcal{M}_1$ and (P_2, Q_2) dominates $\mathcal{M}_2$, then $(P_1 \otimes P_2, Q_1 \otimes Q_2)$ dominates the (adaptively) composed mechanism $\mathcal{M}_1 \circ \mathcal{M}_2$.

3. Dominating Pairs for $\mathrm{ABLQ}_{\mathcal{B}}$

We discuss the dominating pairs for $\mathrm{ABLQ}_{\mathcal{B}}$ for $\mathcal{B} \in \{\mathcal{D}, \mathcal{P}, \mathcal{S}\}$ that will be crucial for establishing our results.

Tightly dominating pair for $\mathrm{ABLQ}_{\mathcal{D}}$. It follows from the standard analysis of the Gaussian mechanism and parallel composition that a tightly dominating pair for $\mathrm{ABLQ}_{\mathcal{D}}$ is the pair $(P_{\mathcal{D}} := \mathcal{N}(1, \sigma^2), Q_{\mathcal{D}} := \mathcal{N}(0, \sigma^2))$, leading to a closed-form expression for $\delta_{\mathcal{D}}(\varepsilon)$.

Proposition 3.1 (Theorem 8 in Balle & Wang (2018))**.** *For all $\varepsilon \geq 0$, it holds that*

$$\delta_{\mathcal{D}}(\varepsilon) = \Phi\left(-\sigma\varepsilon + \tfrac{1}{2\sigma}\right) - e^{\varepsilon}\Phi\left(-\sigma\varepsilon - \tfrac{1}{2\sigma}\right),$$

where $\Phi(\cdot)$ is the cumulative density function (CDF) of the standard normal random variable $\mathcal{N}(0, 1)$.

Tightly dominating pair of $\mathrm{ABLQ}_{\mathcal{P}}$. Zhu et al. (2022) showed[6] that the tightly dominating pair for a single step of

[5] Zhu et al. (2022) define "tightly dominating pair" differently, in a manner that is guaranteed to exist. They additional define the notion of a "worst-case pair", which is a pair of adjacent datasets $\boldsymbol{x} \sim \boldsymbol{x}'$ such that $(\mathcal{M}(\boldsymbol{x}), \mathcal{M}(\boldsymbol{x}'))$ is a tightly dominating pair. Thus, our notion of "tightly dominating pair" refers precisely to the pair $(\mathcal{M}(\boldsymbol{x}), \mathcal{M}(\boldsymbol{x}'))$ for a worst-case adjacent pair $\boldsymbol{x}, \boldsymbol{x}'$. It is also worth noting that our notation for "tightly dominating pairs" is asymmetric as it only considers pairs $\boldsymbol{x} \to_z \boldsymbol{x}'$; the reverse setting is handled implicitly because (P, Q) dominates $(\mathcal{M}(\boldsymbol{x}), \mathcal{M}(\boldsymbol{x}'))$ if and only if (Q, P) dominates $(\mathcal{M}(\boldsymbol{x}'), \mathcal{M}(\boldsymbol{x}))$.

[6] This was implicit in prior work, e.g., (Koskela et al., 2020).

$\text{ABLQ}_{\mathcal{P}}$, a Poisson sub-sampled Gaussian mechanism, is given by the pair $(A = (1-q)\mathcal{N}(0,\sigma^2) + q\mathcal{N}(1,\sigma^2), B = \mathcal{N}(0,\sigma^2))$, where q is the sub-sampling probability of each record, namely $q = b/n$, and in the case where $n = b \cdot T$, we have $q = 1/T$. Since $\text{ABLQ}_{\mathcal{P}}$ is a T-fold composition of this Poisson subsampled Gaussian mechanism, it follows that the tightly dominating pair for $\text{ABLQ}_{\mathcal{P}}$ is $(P_{\mathcal{P}} := A^{\otimes T}, Q_{\mathcal{P}} := B^{\otimes T})$.

The hockey stick divergence $D_{e^\varepsilon}(P_{\mathcal{P}} \| Q_{\mathcal{P}})$ does not have a closed-form expression, but there are privacy accountants based on the methods of Rényi DP (RDP) (Mironov, 2017) as well as *privacy loss distributions (PLD)* (Meiser & Mohammadi, 2018; Sommer et al., 2019), the latter providing numerically accurate algorithms (Koskela et al., 2020; Gopi et al., 2021; Ghazi et al., 2022; Doroshenko et al., 2022), and have been the basis of multiple open-source implementations from both industry and academia including (Prediger & Koskela, 2020; Google's DP Library., 2020; Microsoft., 2021). While Rényi-DP-based accounting provides an upper bound on $\max\{D_{e^\varepsilon}(P_{\mathcal{P}} \| Q_{\mathcal{P}}), D_{e^\varepsilon}(Q_{\mathcal{P}} \| P_{\mathcal{P}})\}$, the PLD-based accounting implementations can provide upper and lower bounds on $\max\{D_{e^\varepsilon}(P_{\mathcal{P}} \| Q_{\mathcal{P}}), D_{e^\varepsilon}(Q_{\mathcal{P}} \| P_{\mathcal{P}})\}$ to high accuracy, as controlled by a certain discretization parameter.

Tightly dominating pair for $\text{ABLQ}_{\mathcal{S}}$? It is not clear which adjacent pair would correspond to a tightly dominating pair for $\text{ABLQ}_{\mathcal{S}}$, and moreover, it is even a priori unclear if one even exists. However, in order to prove *lower bounds* on the privacy parameters, it suffices to consider a specific instantiation of the adaptive query method $\mathcal{A}$, and a particular adjacent pair $\boldsymbol{x} \sim \boldsymbol{x}'$. In particular, we instantiate the query method $\mathcal{A}$ as follows.

Consider the input space $\mathcal{X} = [-1, 1]$, and assume that the query method $\mathcal{A}$ is non-adaptive, and always produces the query $\psi_t(x) = x$. We consider the adjacent datasets:

- $\boldsymbol{x} = (x_1 = -1, \ldots, x_{n-1} = -1, x_n = 1)$, and
- $\boldsymbol{x}' = (x_1 = -1, \ldots, x_{n-1} = -1, x_n = \perp)$.

In this case, when the differing record falls in batch t, then output of the mechanism on all batches $t' \neq t$ is centered at $-b$, and is centered at $-b+2$ (on input $\boldsymbol{x}$) or at $-b+1$ (on input $\boldsymbol{x}'$) on batch t. Thus it follows that the distributions $A = \text{ABLQ}_{\mathcal{S}}(\boldsymbol{x})$ and $B = \text{ABLQ}_{\mathcal{S}}(\boldsymbol{x}')$ are given as:

$$A = \sum_{t=1}^{T} \tfrac{1}{T} \cdot \mathcal{N}(-b \cdot \mathbf{1} + 2e_t, \sigma^2 I),$$
$$B = \sum_{t=1}^{T} \tfrac{1}{T} \cdot \mathcal{N}(-b \cdot \mathbf{1} + e_t, \sigma^2 I),$$

where $\mathbf{1}$ denotes the all-1's vector in $\mathbb{R}^T$ and e_t denotes the tth standard basis vector in $\mathbb{R}^T$. Shifting the distributions by $b \cdot \mathbf{1}$ does not change the hockey stick divergence $D_{e^\varepsilon}(A \| B)$, hence we might as well consider the pair

$$P_{\mathcal{S}} := \sum_{t=1}^{T} \tfrac{1}{T} \cdot \mathcal{N}(2e_t, \sigma^2 I), \quad (1)$$
$$Q_{\mathcal{S}} := \sum_{t=1}^{T} \tfrac{1}{T} \cdot \mathcal{N}(e_t, \sigma^2 I). \quad (2)$$

Thus, $\delta_{\mathcal{S}}(\varepsilon) \geq \max\{D_{e^\varepsilon}(P_{\mathcal{S}} \| Q_{\mathcal{S}}), D_{e^\varepsilon}(Q_{\mathcal{S}} \| P_{\mathcal{S}})\}$. We conjecture that this pair is in fact tightly dominating for $\text{ABLQ}_{\mathcal{S}}$ for all instantiations of query methods $\mathcal{A}$ (including adaptive ones). We elaborate more in Section 4.3.1.

Conjecture 3.2. *The pair $(P_{\mathcal{S}}, Q_{\mathcal{S}})$ tightly dominates $\text{ABLQ}_{\mathcal{S}}$ for all adaptive query methods $\mathcal{A}$.*

The results in this paper do *not* rely on this conjecture being true, as we only use the dominating pair $(P_{\mathcal{S}}, Q_{\mathcal{S}})$ to establish *lower bounds* on $\delta_{\mathcal{S}}(\cdot)$.

4. Privacy Loss Comparisons

4.1. $\text{ABLQ}_{\mathcal{D}}$ vs $\text{ABLQ}_{\mathcal{S}}$

We first note that $\text{ABLQ}_{\mathcal{S}}$ enjoys stronger privacy guarantees than $\text{ABLQ}_{\mathcal{D}}$.

Theorem 4.1. *For all $\sigma, \varepsilon \geq 0$ and $T \geq 1$: $\delta_{\mathcal{S}}(\varepsilon) \leq \delta_{\mathcal{D}}(\varepsilon)$.*

This follows from a standard technique that shuffling cannot degrade the privacy guarantee satisfied by a mechanism. For completeness, we provide a proof in Appendix B.

4.2. $\text{ABLQ}_{\mathcal{D}}$ vs $\text{ABLQ}_{\mathcal{P}}$

We show that $\text{ABLQ}_{\mathcal{D}}$ and $\text{ABLQ}_{\mathcal{P}}$ have incomparable privacy loss. In particular, we show the following.

Theorem 4.2. *For all $\sigma > 0$ and $T > 1$, there exist $\varepsilon_0, \varepsilon_1 \geq 0$ such that,*

(a) $\forall \varepsilon \in [0, \varepsilon_0)$, it holds that $\delta_{\mathcal{D}}(\varepsilon) > \delta_{\mathcal{P}}(\varepsilon)$, and

(b) $\forall \varepsilon > \varepsilon_1$, it holds that $\delta_{\mathcal{D}}(\varepsilon) < \delta_{\mathcal{P}}(\varepsilon)$.

We defer the detailed proof to Appendix C, and provide a proof sketch here. Part (a) is shown by first establishing that the total variation distance (corresponds to $D_1(\cdot\|\cdot)$) between $P_{\mathcal{D}}$ and $Q_{\mathcal{D}}$ is strictly larger than the total variation distance between $P_{\mathcal{P}}$ and $Q_{\mathcal{P}}$ when $T > 1$ and $\sigma > 0$. This implies that, $\delta_{\mathcal{D}}(0) > \delta_{\mathcal{P}}(0)$. By using the continuity of $D_{e^\varepsilon}(\cdot\|\cdot)$ in ε, we conclude the same for all $\varepsilon < \varepsilon_0$.

For part (b), we construct an explicit set E such that $P_{\mathcal{P}}(E) - e^\varepsilon Q_{\mathcal{P}}(E) > \delta_{\mathcal{D}}(\varepsilon)$. In particular, we choose a halfspace $E := \left\{w \in \mathbb{R}^T \middle| \sum_i w_i > (\varepsilon + \log 2 + T \log T)\sigma^2 + \frac{T}{2}\right\}$ and show that $P_{\mathcal{P}}(E) - e^\varepsilon Q_{\mathcal{P}}(E)$ is at least $\frac{1}{2}\Phi\left(-\frac{\varepsilon\sigma}{\sqrt{T}} - \frac{(T\log T + \log 2)\sigma}{\sqrt{T}} - \frac{\sqrt{T}}{2\sigma}\right)$. For large ε, the dominant term is $-\varepsilon\sigma/\sqrt{T}$. On other hand, $\delta_{\mathcal{D}}(\varepsilon)$ is at most $\Phi(-\varepsilon\sigma + \frac{1}{2\sigma})$ (from Proposition 3.1), which has the dominant term $-\varepsilon\sigma$. Since $-\varepsilon\sigma/\sqrt{T}$ decays slower

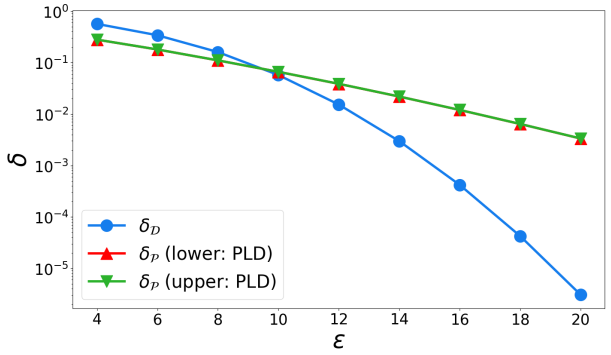

Figure 2. $\delta_{\mathcal{D}}(\varepsilon)$ and $\delta_{\mathcal{P}}(\varepsilon)$ for $\sigma = 0.3$ and $T = 10$.

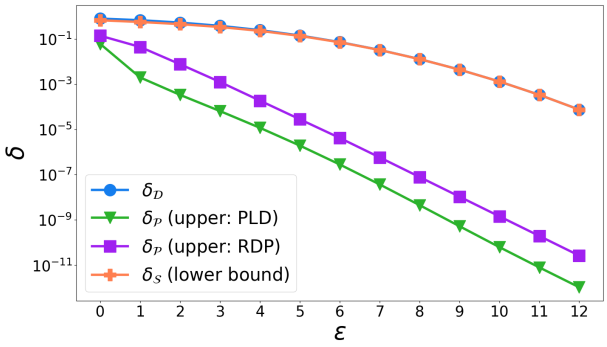

Figure 3. $\delta_{\mathcal{D}}(\varepsilon)$, upper bounds on $\delta_{\mathcal{P}}(\varepsilon)$ and a lower bound on $\delta_{\mathcal{S}}(\varepsilon)$ for varying ε and fixed $\sigma = 0.4$ and $T = 10,000$.

than $-\varepsilon\sigma$, we get that for sufficiently large ε_1, it holds that $\delta_{\mathcal{D}}(\varepsilon) < \delta_{\mathcal{P}}(\varepsilon)$ for all $\varepsilon > \varepsilon_1$.

Even though Theorem 4.2 was proved for some values of ε_0 and ε_1, we conjecture that it holds for $\varepsilon_0 = \varepsilon_1$.

Conjecture 4.3. *Theorem 4.2 holds for* $\varepsilon_0 = \varepsilon_1$.

We do *not* rely on this conjecture being true in the rest of this paper. We provide a numerical example that validates Theorem 4.2 and provides evidence for Conjecture 4.3. In Figure 2, for $\sigma = 0.3$ and $T = 10$, we plot the numerically computed $\delta_{\mathcal{D}}(\varepsilon)$ (using Proposition 3.1), as well as lower and upper bounds on $\delta_{\mathcal{P}}(\varepsilon)$, computed using the open source `dp_accounting` library (Google's DP Library., 2020).

4.3. ABLQ$_{\mathcal{P}}$ vs ABLQ$_{\mathcal{S}}$

From Theorems 4.1 and 4.2, it follows that there exists an ε_1 such that for all $\varepsilon > \varepsilon_1$, it holds that $\delta_{\mathcal{S}}(\varepsilon) \leq \delta_{\mathcal{D}}(\varepsilon) < \delta_{\mathcal{P}}(\varepsilon)$. On the other hand, while we know that $\delta_{\mathcal{D}}(\varepsilon) > \delta_{\mathcal{P}}(\varepsilon)$ for sufficiently small ε, this does not imply anything about the relationship between $\delta_{\mathcal{S}}(\varepsilon)$ and $\delta_{\mathcal{P}}(\varepsilon)$ for small ε.

We demonstrate simple numerical settings where $\delta_{\mathcal{S}}(\varepsilon)$ is significantly larger than $\delta_{\mathcal{P}}(\varepsilon)$. We prove lower bounds on $\delta_{\mathcal{S}}(\varepsilon)$ by constructing specific sets E and using the fact that $\delta_{\mathcal{S}}(\varepsilon) \geq P_{\mathcal{S}}(E) - e^{\varepsilon} Q_{\mathcal{S}}(E)$.

In particular, we consider sets E_C parameterized by C of the form $\{w \in \mathbb{R}^T : \max_t w_t \geq C\}$; note that E_C is the complement of T-fold Cartesian product of the set $(-\infty, C)$. For a single Gaussian distribution $D = \mathcal{N}(\mu, \sigma^2 I)$, we can compute the probability mass of E_C under measure D as:

$$\begin{aligned} D(E_C) &= 1 - D(\mathbb{R}^T \setminus E_C) \\ &= 1 - \textstyle\prod_{t=1}^{T} \Pr_{x \sim \mathcal{N}(\mu_t, \sigma^2)}[x < C] \\ &= 1 - \textstyle\prod_{t=1}^{T} \Phi\left(\frac{C - \mu_t}{\sigma}\right). \end{aligned}$$

In particular, when μ is $\alpha \cdot e_t$ for any standard basis vector e_t, we have $D(E_C) = 1 - \Phi\left(\frac{C-\alpha}{\sigma}\right) \cdot \Phi\left(\frac{C}{\sigma}\right)^{T-1}$. Thus, we have that $P_{\mathcal{S}}(E_C)$ is

$$\begin{aligned} P_{\mathcal{S}}(E_C) &= \textstyle\sum_{t=1}^{T} \frac{1}{T} D_t(E_C) \quad \text{for } D_t = N(2e_t, \sigma^2 I) \\ &= 1 - \Phi\left(\tfrac{C-2}{\sigma}\right) \cdot \Phi\left(\tfrac{C}{\sigma}\right)^{T-1}. \end{aligned}$$

Similarly, we have $Q_{\mathcal{S}}(E_C) = 1 - \Phi\left(\frac{C-1}{\sigma}\right) \cdot \Phi\left(\frac{C}{\sigma}\right)^{T-1}$.

Thus, we use the following lower bound:

$$\delta_{\mathcal{S}}(\varepsilon) \geq \max_{C \in \mathcal{C}} P_{\mathcal{S}}(E_C) - e^{\varepsilon} Q_{\mathcal{S}}(E_C) \tag{3}$$

for any suitable set $\mathcal{C}$ that can be enumerated over. In our experiments described below, we set $\mathcal{C}$ to be the set of all values of C ranging from 0 to 100 in increments of 0.01.

In Figure 3, we set $\sigma = 0.4$ and number of steps $T = 10,000$ and plot $\delta_{\mathcal{D}}(\varepsilon)$, an upper bound on $\delta_{\mathcal{P}}(\varepsilon)$ (obtained using `dp_accounting`) and a lower bound on $\delta_{\mathcal{S}}(\varepsilon)$ as obtained via (3). We find that while $\delta_{\mathcal{P}}(4) \leq 1.18 \cdot 10^{-5}$, $\delta_{\mathcal{S}}(4) \geq 0.226$, that is close to $\delta_{\mathcal{D}}(4) \approx 0.244$. Even $\delta_{\mathcal{S}}(12) \geq 7.5 \cdot 10^{-5}$ is larger than $\delta_{\mathcal{P}}(4)$. While the $(4, 1.2 \cdot 10^{-5})$-DP guarantee of ABLQ$_{\mathcal{P}}$ could have been considered as sufficiently private, ABLQ$_{\mathcal{S}}$ only satisfies much worse privacy guarantees. We provide additional examples in Appendix A.

4.3.1. Intuition for Conjecture 3.2

We attempt to shed some intuition for why ABLQ$_{\mathcal{S}}$ does not provide as much amplification over ABLQ$_{\mathcal{D}}$, compared to ABLQ$_{\mathcal{P}}$, and why we suggest Conjecture 3.2.

For sake of intuition, let's consider the setting where the query method $\mathcal{A}$ always generates the query $\psi_t(x) = x$, and we have two adjacent datasets:

- $\boldsymbol{x} = (x_1 = -L, \ldots, x_{n-1} = -L, x_n = 1)$, and
- $\boldsymbol{x}' = (x_1 = -L, \ldots, x_{n-1} = -L, x_n = \perp)$.

The case of $L > 1$ is not valid, since in this case $|\psi_t(x)| = L > 1$. However, we can still ask how well the privacy

of the nth example is preserved by $\mathrm{ABLQ}_{\mathcal{B}}$, by considering the hockey stick divergence between $\mathrm{ABLQ}_{\mathcal{B}}(\boldsymbol{x})$ and $\mathrm{ABLQ}_{\mathcal{B}}(\boldsymbol{x}')$.

The crucial difference between $\mathrm{ABLQ}_{\mathcal{P}}$ and $\mathrm{ABLQ}_{\mathcal{S}}$ is that the privacy analysis of $\mathrm{ABLQ}_{\mathcal{P}}$ does not depend at all on the non-differing records in the two datasets. In the case of $\mathrm{ABLQ}_{\mathcal{S}}$, we observe that for any fixed σ and T, the hockey stick divergence $D_{e^\varepsilon}(\mathrm{ABLQ}_{\mathcal{S}}(\boldsymbol{x})\|\mathrm{ABLQ}_{\mathcal{S}}(\boldsymbol{x}'))$ approaches $\delta_{\mathcal{D}}(\varepsilon)$ as $L \to \infty$. We sketch this argument intuitively: For any batch S_t that does not contain n, the corresponding $g_t = -bL + e_t$ for $e_t \sim \mathcal{N}(0, \sigma^2)$. Whereas for the batch S_t that contains n, the corresponding $g_t = -(b-1)L + 1 + e_t$ in case of input $\boldsymbol{x}$ or $g_t \sim -(b-1)L + e_t$ in case of input $\boldsymbol{x}'$. As $L \to \infty$, we can identify the batch S_t that contains n with probability approaching 1, thereby not providing any amplification.

In summary, the main differing aspect about $\mathrm{ABLQ}_{\mathcal{S}}$ and $\mathrm{ABLQ}_{\mathcal{P}}$ is that in the former, the non-differing examples can leak information about the location of the differing example in the shuffled order, but it is not the case in the latter. While we sketched an argument that works asymptotically as $L \to \infty$, we see glimpses of it already at $L = 1$.

Our intuitive reasoning behind Conjecture 3.2 is that even in the case of (vector-valued) adaptive query methods, in order to "leak the most information" about the differing record x_i between $\boldsymbol{x}$ and $\boldsymbol{x}'$, it seems natural that the query $\psi_t(x_j)$ should evaluate to $-\psi_t(x_i)$ for all $j \neq i$. If the query method satisfies this condition for all t, then it is easy to show that $(P_{\mathcal{S}}, Q_{\mathcal{S}})$ tightly dominates $(\mathrm{ABLQ}_{\mathcal{S}}(\boldsymbol{x}), \mathrm{ABLQ}_{\mathcal{S}}(\boldsymbol{x}'))$. Conjecture 3.2 then asserts that this is indeed the worst case.

5. Conclusion & Future Directions

We identified significant gaps between the privacy analysis of adaptive batch linear query mechanisms, under the deterministic, Poisson, and shuffle batch samplers. We find that while shuffling always provides better privacy guarantees over deterministic batching, Poisson batch sampling can provide a worse privacy guarantee than even deterministic sampling at large ε. But perhaps most surprisingly, we demonstrate that the amplification guarantees due to shuffle batch sampling can be severely limited compared to the amplification due to Poisson subsampling, in various regimes that could be considered of practical interest.

Several interesting directions remain to be investigated. In our work, we provide a technique to only provide a *lower bound* on the privacy guarantee when using a shuffle batch sampler. It would be interesting to have a tight accounting method for $\mathrm{ABLQ}_{\mathcal{S}}$. A first step towards this could be to establish Conjecture 3.2, which if true, might make numerical accountants for computing the hockey stick divergence possible. While this involves computing a high-dimensional integral, it might be possible to approximate using importance sampling; e.g., such approaches have been attempted for $\mathrm{ABLQ}_{\mathcal{P}}$ (Wang et al., 2023). Also, our approach for analyzing the privacy with shuffle batch sampler is limited to a "single epoch" mechanism, whereas in practice, it is common to use DP-SGD with multiple epochs. Extending our approach to multiple epochs will be interesting.

However, it remains to be seen how the utility of DP-SGD would be affected when we use the correct privacy analysis for $\mathrm{ABLQ}_{\mathcal{S}}$ instead of the analysis for $\mathrm{ABLQ}_{\mathcal{P}}$, which has been used extensively so far and treated as a good "approximation". Alternative approaches such as DP-FTRL (Kairouz et al., 2021; McMahan et al., 2022) that do not rely on amplification might turn out to be better if we instead use the correct analysis for $\mathrm{ABLQ}_{\mathcal{S}}$, in the regimes where such methods are currently reported to under perform compared to DP-SGD.

An important point to note is that the model of shuffle batch sampler we consider here is a simple one. There are various types of data loaders used in practice, which are not necessarily captured by our model. For example `tf.data.Dataset.shuffle` takes in a parameter of buffer size b. It returns a random record among the first b records, and immediately replaces it with the next record ($(b+1)$st in this case), and continues repeating this process. This leads to an asymmetric form of shuffling, when the dataset size exceeds the size of the buffer. Such batch samplers might thus require more careful privacy analysis.

The notion of DP aims to guarantee privacy even in the worst case. For example in the context of DP-SGD, it aims to protect privacy of a single record even when the model trainer and all other records participating in the training are colluding to leak information about this one record. And moreover, releasing the final model is assumed to leak as much information as releasing all intermediate iterates. Such strong adversarial setting might make obtaining good utility to be difficult. Alternative techniques for privacy amplification such as amplification by iteration (Feldman et al., 2018; Altschuler & Talwar, 2022) or through convergence of Langevin dynamics (Chourasia et al., 2021) have been studied, where only the last iterate of the model is released. However, these analyses rely on additional assumptions such as convexity and smoothness of the loss functions. Investigating whether it is possible to relax these assumptions to make amplification by iteration applicable to non-convex models, even if under some assumptions that are applicable to the ones used in practice, is an interesting future direction.

Acknowledgments

We thank the anonymous ICML reviewers for their thoughtful comments and suggestions which have improved the presentation of this paper. We also thank Christian Janos Lebeda, Matthew Regehr, Gautam Kamath, and Thomas Steinke for correspondence about their concurrent work (Lebeda et al., 2024).

Impact Statement

This paper presents work that points out that the particular type of batch sampling used can play a significant role in the privacy analysis of DP-SGD type of algorithms. The impact we see coming from our work is to make practitioners of DP more aware of these gaps. There are many potential societal consequences of DP itself, none which we feel must be specifically highlighted here.

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *CCS*, pp. 308–318, 2016.

Altschuler, J. M. and Talwar, K. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. In *NeurIPS*, 2022.

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. Large-scale differentially private BERT. In *EMNLP (Findings)*, pp. 6481–6491, 2022.

Balle, B. and Wang, Y. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *ICML*, pp. 403–412, 2018.

Balle, B., Berrada, L., De, S., Ghalebikesabi, S., Hayes, J., Pappu, A., Smith, S. L., and Stanforth, R. JAX-Privacy: Algorithms for privacy-preserving machine learning in JAX, 2022. URL http://github.com/google-deepmind/jax_privacy.

Bu, Z., Mao, J., and Xu, S. Scalable and efficient training of large convolutional neural networks with differential privacy. *NeurIPS*, pp. 38305–38318, 2022.

Chen, D., Orekondy, T., and Fritz, M. GS-WGAN: a gradient-sanitized approach for learning differentially private generators. In *NeurIPS*, pp. 12673–12684, 2020.

Chourasia, R., Ye, J., and Shokri, R. Differential privacy dynamics of Langevin diffusion and noisy gradient descent. In *NeurIPS*, pp. 14771–14781, 2021.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. *arXiv*, 2204.13650, 2022.

Denison, C., Ghazi, B., Kamath, P., Kumar, R., Manurangsi, P., Narra, K. G., Sinha, A., Varadarajan, A. V., and Zhang, C. Private ad modeling with DP-SGD. In *AdKDD*, 2023.

Dockhorn, T., Cao, T., Vahdat, A., and Kreis, K. Differentially private diffusion models. *arXiv*, 2210.09929, 2022.

Doroshenko, V., Ghazi, B., Kamath, P., Kumar, R., and Manurangsi, P. Connect the dots: Tighter discrete approximations of privacy loss distributions. *PoPETS*, 2022(4): 552–570, 2022.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pp. 2468–2479, 2019.

Fang, L., Du, B., and Wu, C. Differentially private recommender system with variational autoencoders. *Knowledge-Based Systems*, 250:109044, 2022.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. Privacy amplification by iteration. In *FOCS*, pp. 521–532, 2018.

Feldman, V., McMillan, A., and Talwar, K. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *FOCS*, pp. 954–964, 2021.

Feldman, V., McMillan, A., and Talwar, K. Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. In *SODA*, pp. 4966–4981, 2023.

Ghazi, B., Kamath, P., Kumar, R., and Manurangsi, P. Faster privacy accounting via evolving discretization. In *ICML*, pp. 7470–7483, 2022.

Google Colab. URL https://colab.research.google.com/.

Google's DP Library. DP Accounting Library. https://github.com/google/differential-privacy/tree/main/python/dp_accounting, 2020.

Gopi, S., Lee, Y. T., and Wutschitz, L. Numerical composition of differential privacy. In *NeurIPS*, pp. 11631–11642, 2021.

He, J., Li, X., Yu, D., Zhang, H., Kulkarni, J., Lee, Y. T., Backurs, A., Yu, N., and Bian, J. Exploring the limits of differentially private deep learning with group-wise clipping. In *ICLR*, 2023.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *ICML*, pp. 5213–5225, 2021.

Klause, H., Ziller, A., Rueckert, D., Hammernik, K., and Kaissis, G. Differentially private training of residual networks with scale normalisation. *arXiv*, 2203.00324, 2022.

Koskela, A., Jälkö, J., and Honkela, A. Computing tight differential privacy guarantees using FFT. In *AISTATS*, pp. 2560–2569, 2020.

Lebeda, C. J., Regehr, M., Kamath, G., and Steinke, T. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition, 2024. URL https://arxiv.org/abs/2405.20769.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *ICLR*, 2022.

McMahan, B., Rush, K., and Thakurta, A. G. Private online prefix sums via optimal matrix factorizations. *arXiv*, 2202.08312, 2022.

Meiser, S. and Mohammadi, E. Tight on budget? Tight bounds for r-fold approximate differential privacy. In *CCS*, pp. 247–264, 2018.

Microsoft. A fast algorithm to optimally compose privacy guarantees of differentially private (DP) mechanisms to arbitrary accuracy. https://github.com/microsoft/prv_accountant, 2021.

Mironov, I. Rényi differential privacy. In *CSF*, pp. 263–275, 2017.

Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. In *AAAI*, pp. 9312–9321, 2021.

Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. G. How to dp-fy ML: A practical guide to machine learning with differential privacy. *J. Artif. Intell. Res.*, 77:1113–1201, 2023.

Prediger, L. and Koskela, A. Code for computing tight guarantees for differential privacy. https://github.com/DPBayes/PLD-Accountant, 2020.

Sommer, D. M., Meiser, S., and Mohammadi, E. Privacy loss classes: The central limit theorem in differential privacy. *PoPETS*, 2019(2):245–269, 2019.

Tensorflow Privacy, a. URL https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy.

Tensorflow Privacy, b. URL https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy/compute_dp_sgd_privacy_statement. Note about compute_dp_sgd_privacy_statement.

Torkzadehmahani, R., Kairouz, P., and Paten, B. DP-CGAN: Differentially private synthetic data and label generation. In *CVPR Workshops*, 2019.

Tramer, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *ICLR*, 2021.

Vadhan, S. *The Complexity of Differential Privacy*. Springer, 2017.

Wang, J. T., Mahloujifar, S., Wu, T., Jia, R., and Mittal, P. A randomized approach to tight privacy accounting. In *NeurIPS*, 2023.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. Opacus: User-friendly differential privacy library in PyTorch. *arXiv*, 2109.12298, 2021.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. Differentially private fine-tuning of language models. In *ICLR*, 2022.

Zeighami, S., Ahuja, R., Ghinita, G., and Shahabi, C. A neural database for differentially private spatial range queries. In *VLDB*, pp. 1066–1078, 2022.

Zhu, Y., Dong, J., and Wang, Y. Optimal accounting of differential privacy via characteristic function. In *AISTATS*, pp. 4782–4817, 2022.

Ziller, A., Usynin, D., Braren, R., Makowski, M., Rueckert, D., and Kaissis, G. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1):13524, 2021.

A. Additional Evaluations

A.1. ε vs. σ for fixed δ and T

We first plot ε against σ for fixed δ and T. We compute upper bounds on $\varepsilon_{\mathcal{P}}(\delta)$ using Rényi DP (RDP) as well as using privacy loss distributions (PLD). These accounting methods are provided in the open source Google `dp_accounting` library (Google's DP Library., 2020).

In particular we consistently find that for small values of σ, $\text{ABLQ}_{\mathcal{S}}$ provides almost no improvement over $\text{ABLQ}_{\mathcal{D}}$, and has $\varepsilon_{\mathcal{S}}$ that is significantly larger than $\varepsilon_{\mathcal{P}}$.

- In Figure 4, we set $\delta = 10^{-6}$ and number of steps $T = 100,000$. In particular, for $\sigma = 0.4$, we find that $\varepsilon_{\mathcal{P}}(\delta) \leq 3$ (as per PLD accounting) and $\varepsilon_{\mathcal{P}}(\delta) \leq 4.71$ (as per RDP accounting), whereas on the other hand, $\varepsilon_{\mathcal{S}}(\delta) \geq 14.45$, which is very close to $\varepsilon_{\mathcal{D}}(\delta)$. For $\sigma = 1.3$, we find that $\varepsilon_{\mathcal{P}}(\delta) < 0.01$ (as per PLD accounting), whereas, $\varepsilon_{\mathcal{S}}(\delta) > 0.029$.

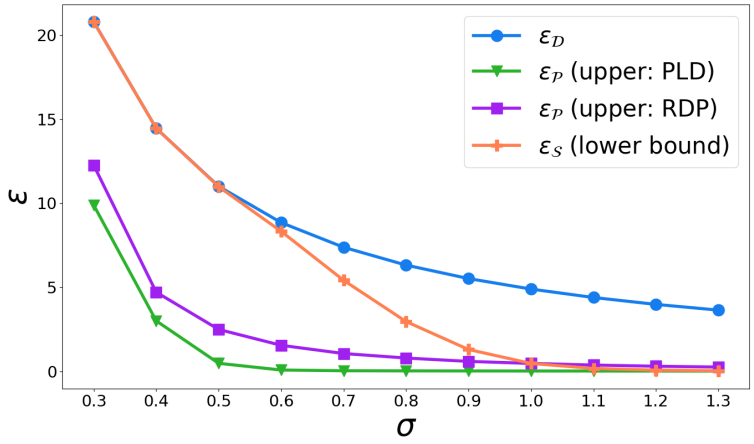

Figure 4. $\varepsilon_{\mathcal{D}}(\delta)$, upper bounds on $\varepsilon_{\mathcal{P}}(\delta)$ and a lower bound on $\varepsilon_{\mathcal{S}}(\delta)$ for varying σ and fixed $\delta = 10^{-6}$ and $T = 10,000$.

- In Figure 5, we set $\delta = 10^{-5}$ and number of steps $T = 1000$. In particular, for $\sigma = 0.7$, $\varepsilon_{\mathcal{P}}(\delta) \leq 0.61$ (as per PLD accounting) and $\varepsilon_{\mathcal{P}}(\delta) \leq 1.64$ (as per RDP accounting), whereas on the other hand, $\varepsilon_{\mathcal{S}}(\delta) \geq 6.528$ and $\varepsilon_{\mathcal{D}}(\delta) \approx 6.652$. For $\sigma = 1.3$, we find that $\varepsilon_{\mathcal{P}}(\delta) < 0.092$ (as per PLD accounting), whereas, $\varepsilon_{\mathcal{S}}(\delta) > 0.83$.

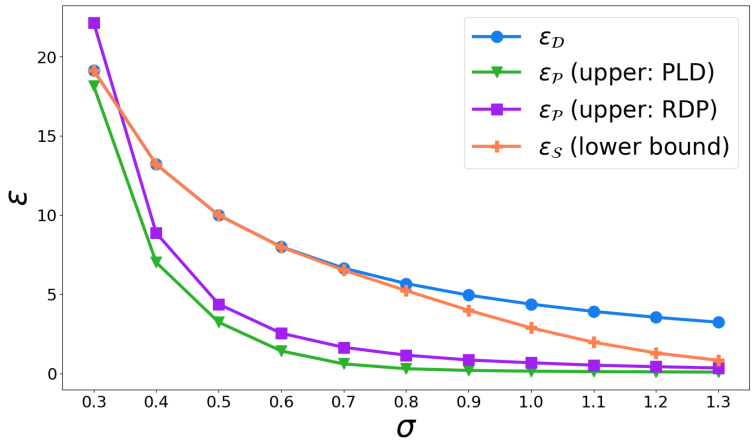

Figure 5. $\varepsilon_{\mathcal{D}}(\delta)$, upper bounds on $\varepsilon_{\mathcal{P}}(\delta)$ and a lower bound on $\varepsilon_{\mathcal{S}}(\delta)$ for varying σ and fixed $\delta = 10^{-5}$ and $T = 1000$.

A.2. δ vs. ε for fixed σ and T

Next, we plot δ against ε for fixed σ and T. We compute upper bounds on $\delta_{\mathcal{P}}(\varepsilon)$ using Rényi DP (RDP) as well as using privacy loss distributions (PLD).

In particular when σ is closer to 1.0, we find that our lower bound on $\delta_{\mathcal{S}}(\varepsilon)$ is distinctly smaller than $\delta_{\mathcal{D}}(\varepsilon)$, but still significantly larger than $\delta_{\mathcal{P}}(\varepsilon)$.

- In Figure 6, we set $\sigma = 0.8$ and number of steps $T = 1000$. In particular, while $\delta_{\mathcal{P}}(1) \leq 9.873 \cdot 10^{-9}$ (as per PLD accounting) and $\delta_{\mathcal{P}}(1) \leq 3.346 \cdot 10^{-5}$ (as per RDP accounting), we have that $\delta_{\mathcal{S}}(1) \geq 0.018$ and $\delta_{\mathcal{S}}(4) \geq 1.6 \cdot 10^{-4}$. For $\varepsilon = 4$, we find the upper bound using PLD accounting to be larger than the upper bound using Rényi DP accounting. This is attributable to the numerical instability in PLD accounting when δ is very small.

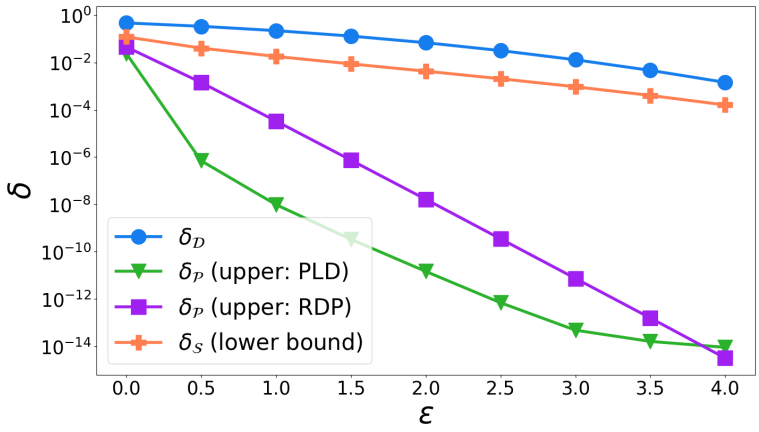

Figure 6. $\delta_{\mathcal{D}}(\varepsilon)$, upper bounds on $\delta_{\mathcal{P}}(\varepsilon)$ and a lower bound on $\delta_{\mathcal{S}}(\varepsilon)$ for varying ε and fixed $\sigma = 0.8$ and $T = 1000$.

- In Figure 7, we set $\sigma = 1.0$ and number of steps $T = 1000$. In particular, while $\delta_{\mathcal{P}}(1) \leq 2.06 \cdot 10^{-10}$ (as per PLD accounting) and $\delta_{\mathcal{P}}(1) \leq 8.45 \cdot 10^{-5}$ (as per RDP accounting), we have that $\delta_{\mathcal{S}}(1) \geq 0.004$ and $\delta_{\mathcal{S}}(4) \geq 4.38 \cdot 10^{-7}$ (last one not shown in plot).
  For $\varepsilon > 1.0$, we find the upper bound using PLD accounting appears to not decrease as much, which could be due to the numerical instability in PLD accounting when δ is very small.

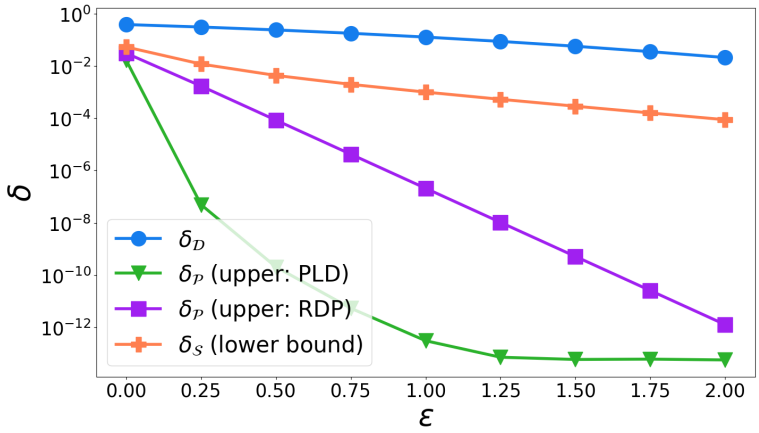

Figure 7. $\delta_{\mathcal{D}}(\varepsilon)$, upper bounds on $\delta_{\mathcal{P}}(\varepsilon)$ and a lower bound on $\delta_{\mathcal{S}}(\varepsilon)$ for varying ε and fixed $\sigma = 1.0$ and $T = 1000$.

B. Proof of Theorem 4.1: ($\text{ABLQ}_{\mathcal{D}}$ vs $\text{ABLQ}_{\mathcal{S}}$)

We use the joint convexity property of hockey stick divergence. While this is standard, we include a proof for completeness.

Lemma B.1 (Joint Convexity of Hockey Stick Divergence)**.** *Given two mixture distributions* $P = \sum_{i=1}^{m} \alpha_i P_i$ *and* $Q = \sum_{i=1}^{m} \alpha_i Q_i$*, it holds for all* $\varepsilon \in \mathbb{R}$ *that* $D_{e^\varepsilon}(P\|Q) \leq \sum_i \alpha_i D_{e^\varepsilon}(P_i\|Q_i)$ *.*

Proof. We have that

$$\begin{aligned} D_{e^\varepsilon}(P\|Q) &= \sup_E \{P(E) - e^\varepsilon Q(E)\} \\ &= \sup_E \left\{ \sum_{i=1}^{m} \alpha_i \left(P_i(E) - e^\varepsilon Q_i(E)\right) \right\} \\ &\leq \sum_{i=1}^{m} \alpha_i \cdot \sup_E \{P_i(E) - e^\varepsilon Q_i(E)\} = \sum_{i=1}^{m} \alpha_i D_{e^\varepsilon}(P_i\|Q_i)\,. \end{aligned}$$

□

Thus, it follows that shuffling the dataset first cannot degrade the privacy guarantee of *any* mechanism as shown below.

Lemma B.2. *Fix a mechanism* $\mathcal{M} : \mathcal{X}^* \to \Delta_{\mathcal{O}}$*, and let* $\mathcal{M}_{\mathcal{S}}$ *be defined as* $\mathcal{M}_{\mathcal{S}}(\boldsymbol{x}) := \mathcal{M}(\boldsymbol{x}_\pi)$ *for a random permutation* π *over* $[n]$ *where* $\boldsymbol{x}_\pi := (x_{\pi(1)}, \ldots, x_{\pi(n)})$*. Then, if* $\mathcal{M}$ *satisfies* (ε, δ)-DP*, then* $\mathcal{M}_{\mathcal{S}}$ *also satisfies* (ε, δ)-DP*.*

Proof. Consider any adjacent pair of dataset $\boldsymbol{x} \sim \boldsymbol{x}'$. For any permutation π over $[n]$, let $P_\pi := \mathcal{M}(\boldsymbol{x}_\pi)$ and $Q_\pi := \mathcal{M}(\boldsymbol{x}'_\pi)$, and let $P = \mathcal{M}_{\mathcal{S}}(\boldsymbol{x})$ and $Q = \mathcal{M}_{\mathcal{S}}(\boldsymbol{x}')$. It is easy to see that

$$P = \sum_\pi \tfrac{1}{n!} \cdot P_\pi \qquad \text{and} \qquad Q = \sum_\pi \tfrac{1}{n!} \cdot Q_\pi\,.$$

Since $\mathcal{M}$ satisfies (ε, δ)-DP it follows that $D_{e^\varepsilon}(P_\pi\|Q_\pi) \leq \delta$ for all permutations π. Thus, from Lemma B.1, it follows that $D_{e^\varepsilon}(P\|Q) \leq \sum_\pi \frac{1}{n!} D_{e^\varepsilon}(P_\pi\|Q_\pi) \leq \delta$. Hence $\mathcal{M}_{\mathcal{S}}$ also satisfies (ε, δ)-DP. □

Proof of Theorem 4.1. The proof follows by observing that if we choose $\mathcal{M} = \mathrm{ABLQ}_{\mathcal{D}}$ in Lemma B.2, then $\mathrm{ABLQ}_{\mathcal{S}}$ is precisely the corresponding mechanism $\mathcal{M}_{\mathcal{S}}$. □

C. Proof of Theorem 4.2 ($\mathrm{ABLQ}_{\mathcal{D}}$ vs $\mathrm{ABLQ}_{\mathcal{P}}$)

We first state and prove some intermediate statements required for the proof of Theorem 4.2. We use the Gaussian measure of a halfspace.

Proposition C.1. *For* $P = \mathcal{N}(\mu, \sigma^2 I)$ *and the set* $E := \{w \in \mathbb{R}^d : a^\top w - b \geq 0\}$*, it holds that* $P(E) = \Phi\left(\frac{a^\top \mu - b}{\sigma \|a\|_2}\right)$ *.*

Proposition C.2. *For all* $T \in \mathbb{N}$ *and distributions* A, B*, it holds that* $D_1(A^{\otimes T}\|B^{\otimes T}) \leq 1 - (1 - D_1(A\|B))^T$*. And hence* $D_1(A^{\otimes T}\|B^{\otimes T}) \leq T \cdot D_1(A\|B)$ *and equality can occur only if* $T = 1$ *or* $D_1(A\|B) = 0$*.*

Proof. Recall that $D_1(A\|B)$ is the total variation distance between A and B, which has the following characterization $\inf_{(X,Y)} \Pr[X \neq Y]$ where (X, Y) is a coupling such that $X \sim A, Y \sim B$. Given any coupling (X, Y) for A, B, we construct a coupling $((X_1, \ldots, X_T), (Y_1, \ldots, Y_T))$ of $A^{\otimes T}, B^{\otimes T}$ by sampling (X_i, Y_i) independently according to the coupling (X, Y). From this, we have

$$\begin{aligned} &\Pr[(X_1, \ldots, X_T) \neq (Y_1, \ldots, Y_T)] \\ &= 1 - \Pr[(X_1, \ldots, X_T) = (Y_1, \ldots, Y_T)] \\ &= 1 - \Pr[X = Y]^T. \end{aligned}$$

By taking the infimum over all (X, Y) such that $X \sim A, Y \sim B$, we arrive at the desired bound. □

We also note that a simple observation that for all P, Q, the hockey stick divergence $D_{e^\varepsilon}(P\|Q)$ is a 1-Lipschitz in e^ε.

Proposition C.3. *For* $\varepsilon_1 < \varepsilon_2$*, it holds that* $D_{e^{\varepsilon_1}}(P\|Q) - D_{e^{\varepsilon_2}}(P\|Q) \leq e^{\varepsilon_2} - e^{\varepsilon_1}$*.*

Proof. We have that

$$\begin{aligned} D_{e^{\varepsilon_1}}(P\|Q) - D_{e^{\varepsilon_2}}(P\|Q) &= \sup_E \left[P(E) - e^{\varepsilon_1}Q(E)\right] - \sup_{E'}\left[P(E') - e^{\varepsilon_2}Q(E')\right] \\ &\le \sup_E \left[P(E) - e^{\varepsilon_1}Q(E) - P(E) + e^{\varepsilon_2}Q(E)\right] \\ &= (e^{\varepsilon_2} - e^{\varepsilon_1}) \cdot \sup_E Q(E) \\ &= e^{\varepsilon_2} - e^{\varepsilon_1}\,. \end{aligned}$$

□

Proof of Theorem 4.2. We prove each part as follows:

For part (a), first we consider the case of $\varepsilon = 0$. In this case, $D_{e^\varepsilon}(P\|Q)$ is simply the total variation distance between P and Q. Recall that $P_{\mathcal{P}} = A^{\otimes T}$ and $Q_{\mathcal{P}} = B^{\otimes T}$, where $A = (1 - \frac{1}{T})Q_{\mathcal{D}} + \frac{1}{T}P_{\mathcal{D}}$ and $B = Q_{\mathcal{D}}$. Observe that $D_1(A\|B) = \frac{1}{T} \cdot D_1(P_{\mathcal{D}}\|Q_{\mathcal{D}})$. Thus, we have that

$$\begin{aligned} D_1(P_{\mathcal{P}}\|Q_{\mathcal{P}}) &= D_1(A^{\otimes T}\|B^{\otimes T}) \\ &< T \cdot D_1(A\|B) \qquad \text{(Proposition C.2)} \\ &= T \cdot \frac{1}{T} \cdot D_1(P_{\mathcal{D}}\|Q_{\mathcal{D}}) \\ &= D_1(P_{\mathcal{D}}\|Q_{\mathcal{D}})\,. \end{aligned}$$

Note that the inequality is strict for $T > 1$. Since $D_{e^\varepsilon}(P\|Q)$ is continuous in ε (see Proposition C.3), there exists some $\varepsilon_0 > 0$, such that for all $\varepsilon \in [0, \varepsilon_0)$, $\delta_{\mathcal{D}}(\varepsilon) > \delta_{\mathcal{P}}(\varepsilon)$.

For part (b), we construct an explicit bad event $E \subseteq \mathbb{R}^T$ such that $P_{\mathcal{P}}(E) - e^\varepsilon Q_{\mathcal{P}}(E) > D_{e^\varepsilon}(P_{\mathcal{D}}\|Q_{\mathcal{D}})$. In particular, we consider:

$$E := \left\{w \in \mathbb{R}^T \,\middle|\, \textstyle\sum_i w_i > (\varepsilon + \log 2 + T\log T)\sigma^2 + \frac{T}{2}\right\}.$$

The choice of E is such that,

$$\begin{aligned} \log \tfrac{P_{\mathcal{P}}(w)}{Q_{\mathcal{P}}(w)} &= \textstyle\sum_{t=1}^T \log \frac{A(w_t)}{B(w_t)} \\ &= \textstyle\sum_{t=1}^T \log\left(1 - \frac{1}{T} + \frac{1}{T} \cdot e^{\frac{2w_t - 1}{2\sigma^2}}\right) \\ &\ge \textstyle\sum_{t=1}^T \left(\frac{2w_t - 1}{2\sigma^2} - \log T\right) \\ &\ge \tfrac{\sum_{t=1}^T w_t}{\sigma^2} - T\log T - \tfrac{T}{2\sigma^2} \\ &\ge \varepsilon + \log 2 \qquad \text{for all } w \in E\,. \end{aligned}$$

Hence it follows that $\log \frac{P_{\mathcal{P}}(E)}{Q_{\mathcal{P}}(E)} \ge \varepsilon + \log 2$, or equivalently, $P_{\mathcal{P}}(E) \ge 2e^\varepsilon Q_{\mathcal{P}}(E)$. This implies that $D_{e^\varepsilon}(P_{\mathcal{P}}\|Q_{\mathcal{P}}) \ge \frac{1}{2}P_{\mathcal{P}}(E)$.

Next, we obtain a lower bound on $P_{\mathcal{P}}(E)$. For $N_\mu := \mathcal{N}(\mu, \sigma^2 I)$ and $p_k = \frac{1}{T^{-k}}(1 - \frac{1}{T})^{T-k}$, it holds that

$$\begin{aligned} P_{\mathcal{P}}(E) &= \textstyle\sum_{\mu \in \{0,1\}^T} p_{\|\mu\|_1} N_\mu(E) \;\ge\; N_{\mathbf{0}}(E) \\ &= \Phi\left(-\tfrac{\varepsilon\sigma}{\sqrt{T}} - \tfrac{(T\log T + \log 2)\sigma}{\sqrt{T}} - \tfrac{\sqrt{T}}{2\sigma}\right), \end{aligned} \tag{4}$$

where we use Proposition C.1 in the last two steps. On the other hand, we have from Proposition 3.1 that

$$\begin{aligned} \delta_{\mathcal{D}}(\varepsilon) &= \Phi\left(-\varepsilon\sigma + \tfrac{1}{2\sigma}\right) - e^\varepsilon \Phi\left(-\varepsilon\sigma - \tfrac{1}{2\sigma}\right) \\ &\le \Phi\left(-\varepsilon\sigma + \tfrac{1}{2\sigma}\right). \end{aligned} \tag{5}$$

There exists a sufficiently large ε_1 such that

$$\begin{aligned}
\delta_{\mathcal{P}}(\varepsilon) &\geq D_{e^\varepsilon}(P_{\mathcal{P}} \| Q_{\mathcal{P}}) \\
&\geq \tfrac{1}{2} P_{\mathcal{P}}(E) \\
&\geq \tfrac{1}{2}\Phi\left(-\tfrac{\varepsilon\sigma}{\sqrt{T}} - \tfrac{(T\log T + \log 2)\sigma}{\sqrt{T}} - \tfrac{\sqrt{T}}{2\sigma}\right) \qquad (6) \\
&\geq \Phi\left(-\varepsilon\sigma + \tfrac{1}{2\sigma}\right) \qquad \text{(for } \varepsilon > \varepsilon_1\text{)} \qquad (7) \\
&\geq \delta_{\mathcal{D}}(\varepsilon)\,,
\end{aligned}$$

by noting that for large ε the most significant term inside $\Phi(\cdot)$ in (6) is $-\varepsilon\sigma/\sqrt{T}$, whereas in (7) the most significant term inside $\Phi(\cdot)$ is $-\varepsilon\sigma$, which decreases much faster as $\varepsilon \to \infty$, for a fixed $T > 1$ and $\sigma > 0$. □