Scalable DP-SGD: Shuffling vs Poisson Subsampling

Lynn Chua Badih Ghazi Pritish Kamath Ravi Kumar

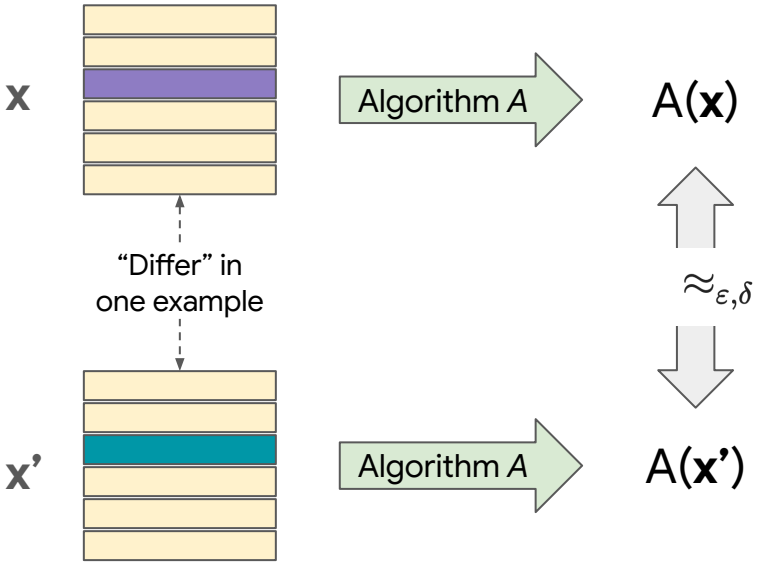
Pasin Manurangsi Amer Sinha Chiyuan Zhang

Google Research

NEURAL INFORMATION PROCESSING SYSTEMS

NeurIPS 2024

Differential Privacy



(ε, δ) -Differential Privacy (DP) [Dwork et al.'06]
For all "adjacent" **x**, **x'** and for all E,

$$\Pr[A(\mathbf{x}) \in E] \leq e^{\varepsilon} \cdot \Pr[A(\mathbf{x}') \in E] + \delta$$

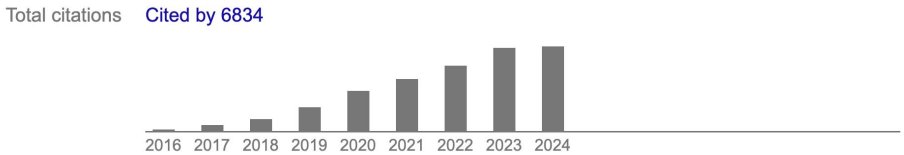
Training models with DP-SGD

A preliminary version of this paper appears in the proceedings of the *23rd ACM Conference on Computer and Communications Security (CCS 2016)*. This is a full version.

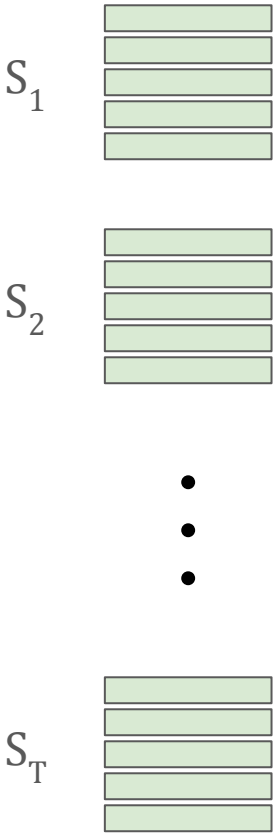
Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi* Andy Chu* Ian Goodfellow†
H. Brendan McMahan* Ilya Mironov* Kunal Talwar*
Li Zhang*



Training models with SGD (mini-batch version)



Starting point:

Differentiable loss $f_w : \mathcal{X} \to \mathbb{R}$

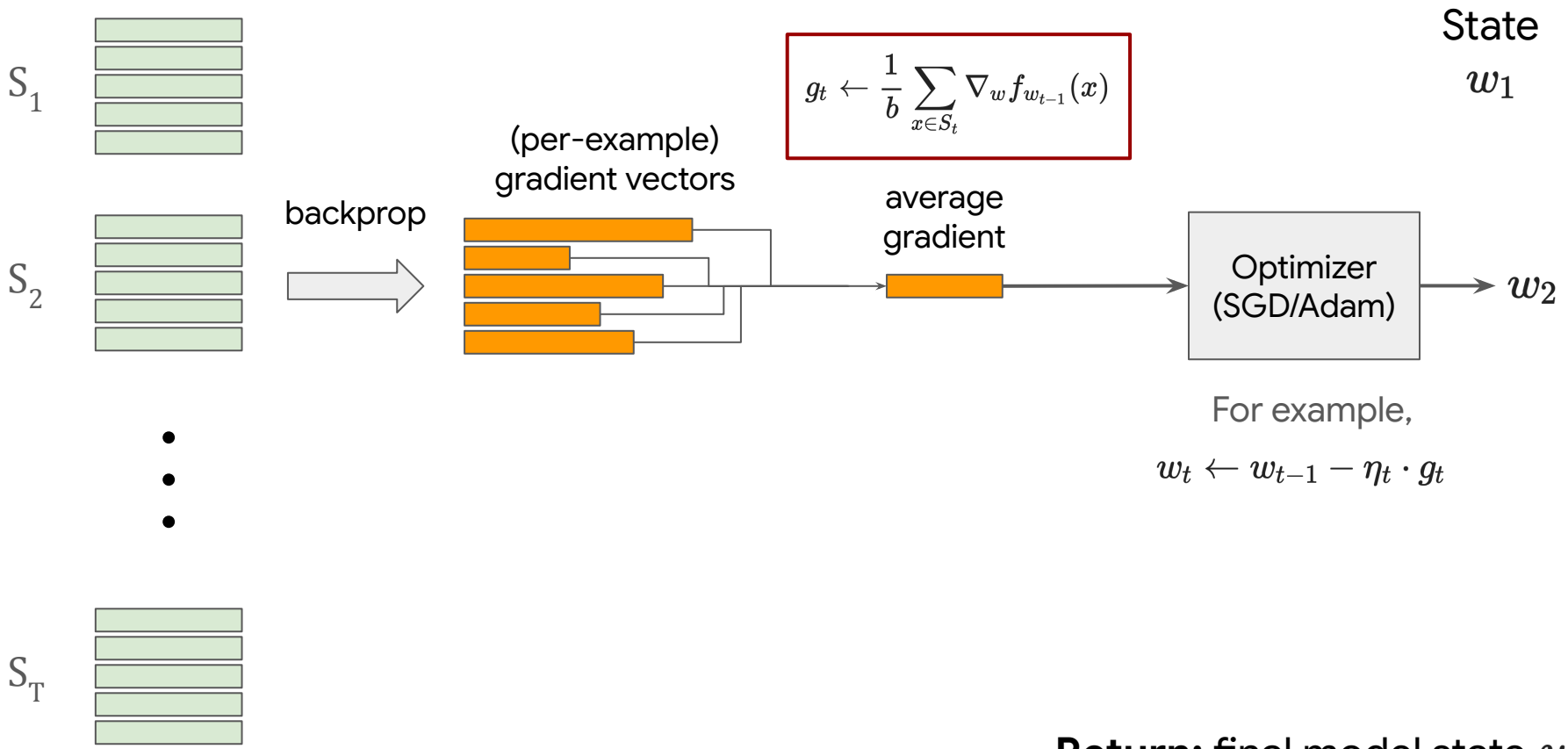
Initial state w_0

Optimizer E.g. : $w_t \leftarrow w_{t-1} - \eta_t g_t$
(SGD, Adam, etc.)

Dataset with n training examples:

- Arrange into batches $S_1, \ldots, S_T$ each of size b
- Assume **single epoch**: $n = b \cdot T$

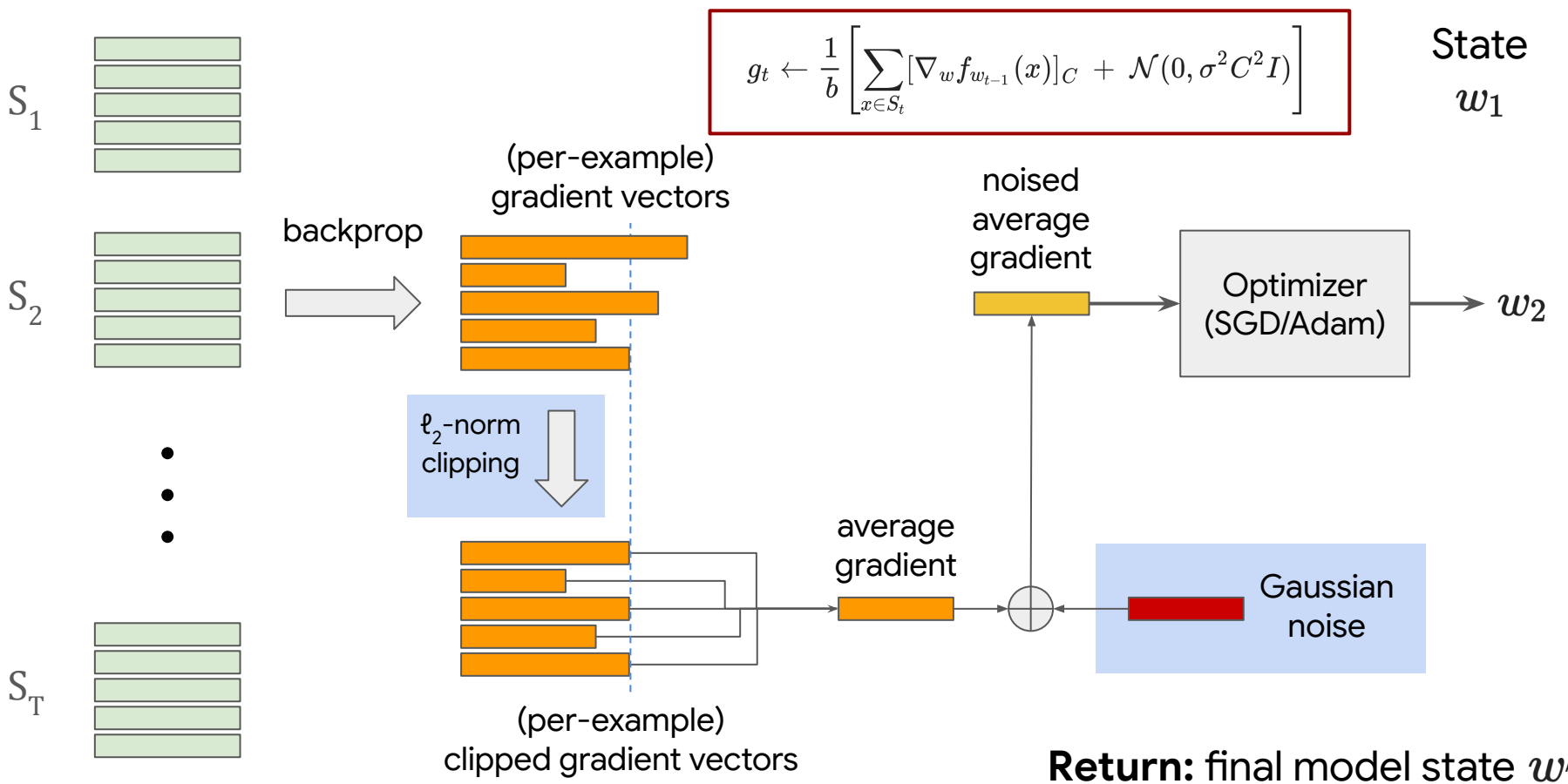
Training models with SGD (mini-batch version)



Return: final model state w_T

Training models with DP-SGD

$$g_t \leftarrow \frac{1}{b}\left[\sum_{x \in S_t}[\nabla_w f_{w_{t-1}}(x)]_C + \mathcal{N}(0, \sigma^2 C^2 I)\right]$$



Return: final model state w_T

Batch Samplers

Construct mini-batches of data each of size b (assume $n = b.T$)

$$(S_1, \ldots, S_T) \leftarrow \mathcal{B}_b(n)$$

Batch Sampler
$\mathcal{B}$

Deterministic
$\mathcal{D}$

Batches of size b in fixed deterministic order

- For t = 1, … , T : $S_t = \{(t-1)b + 1, \ldots, tb\}$

Shuffle (Persistent/Dynamic)
$\mathcal{S}$

Batches of size b in random shuffled order for random permutation π over [n]

- For t = 1, … , T : $S_t = \{\pi((t-1)b + 1), \ldots, \pi(tb)\}$
- π can be fixed (Persistent Shuffle) or vary (Dynamic Shuffle) across epochs

Poisson Subsample
$\mathcal{P}$

Each batch independent with ***expected size*** b; include each coordinate w.p. b / n

- For t = 1, … , T : set $S_t \leftarrow \emptyset$
  - For i = 1, … , n : $S_t \leftarrow \begin{cases} S_t \cup \{i\} & \text{w.p. } \frac{b}{n} \\ S_t & \text{w.p. } 1 - \frac{b}{n} \end{cases}$

Implementation vs Privacy Analysis?

(Shuffling)

[Abadi et al. '16]

We perform the computation in batches, then group several batches into a lot for adding noise. 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.

As is common in the literature, we normalize the running

compute_dp_sgd_privacy_statement

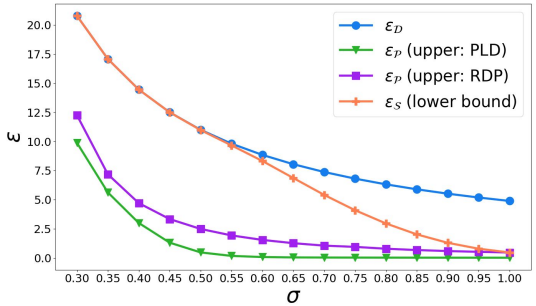
```
DP-SGD performed over 10000 examples with 64 examples per iteration, noise
multiplier 2.0 for 5.0 epochs with microbatching, and at most 3 examples per
user.

This privacy guarantee protects the release of all model checkpoints in addition
to the final model.

Example-level DP with add-or-remove-one adjacency at delta = 1e-06 computed with
PLD accounting:
    Epsilon with each example occurring once per epoch:        12.595
    Epsilon assuming Poisson sampling (*):                      1.199

User-level DP epsilon computation is not supported for PLD accounting at this
time. Use RDP accounting to obtain user-level DP guarantees.

(*) Poisson sampling is not usually done in training pipelines, but assuming
that the data was randomly shuffled, it is believed that the actual epsilon
should be closer to this value than the conservative assumption of an arbitrary
data order.
```

(Poisson Subsampling)

PyTorch Opacus [Yousefpour et al. '21]

Poisson sampling. Opacus also supports uniform sampling of batches (also called Poisson sampling): each data point is independently added to the batch with probability equal to the sampling rate. Poisson sampling is necessary in some analyses of DP-SGD [14].

How Private are DP-SGD Implementations? [Chua et al. '24]

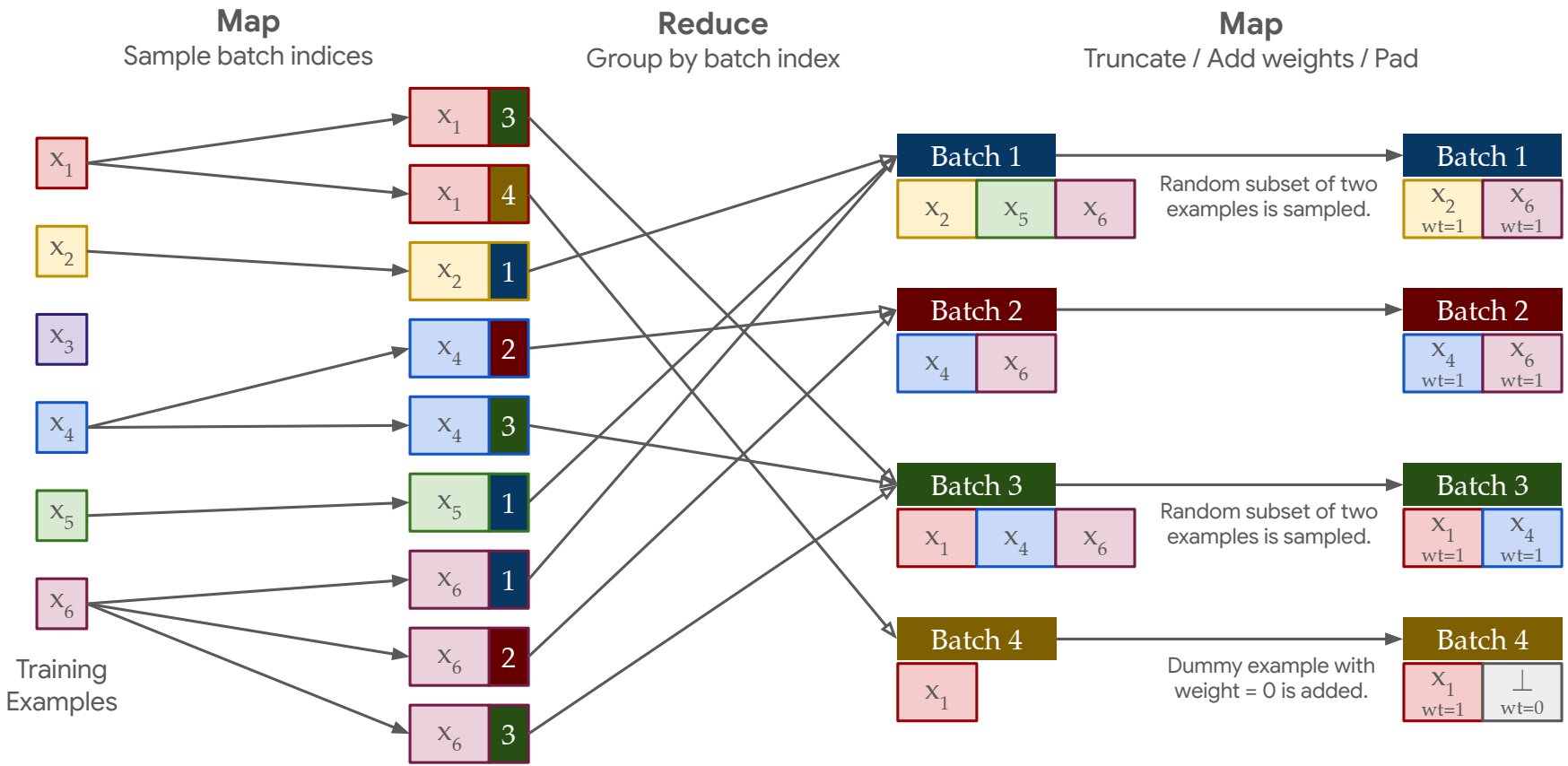


Poisson subsampling version. Our result shows that there can be a substantial gap between the privacy analysis when using the two types of batch sampling, and thus advises caution in reporting privacy parameters for DP-SGD.

This work:

- Extend privacy analysis lower bound for shuffling to multiple epochs
- Compare Poisson subsampling and shuffling with the correct implementation and privacy analysis

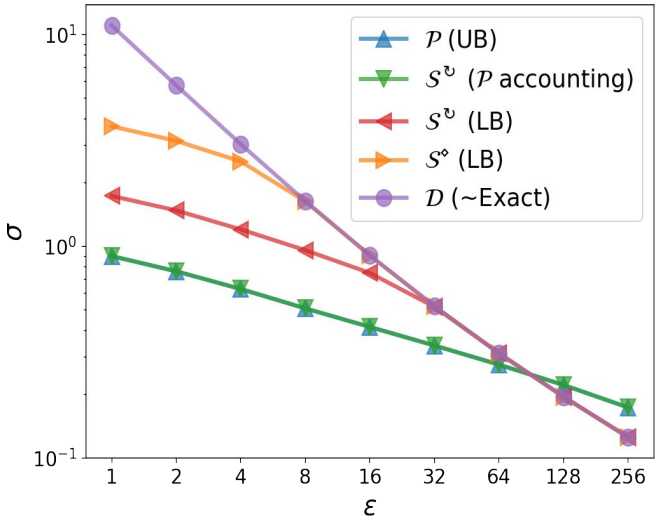
Implementing Poisson subsampling at scale



Results

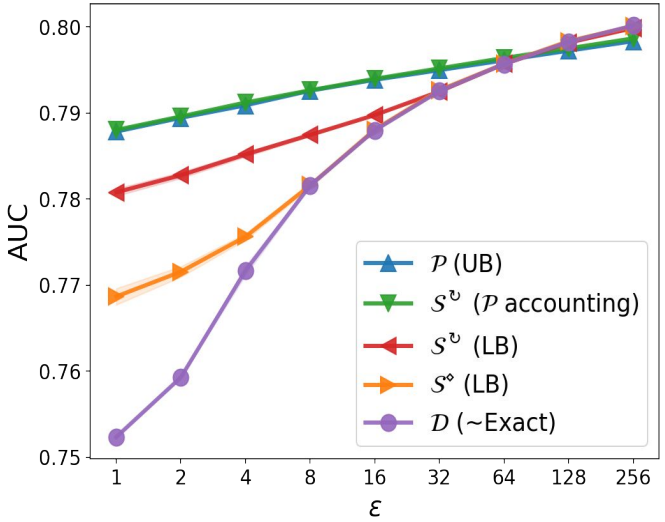
DP-SGD training on Ads dataset with 5 epochs and varying ε

Privacy analysis



- New lower bounds for persistent and dynamic shuffling, for **multiple** epochs

Model utility



- Poisson subsampling has similar utility to Shuffling at the same noise level

Poisson subsampling has better privacy-utility trade-off than shuffling in many practical regimes

Summary

- Poisson subsampling is a viable option for implementing DP-SGD *at scale*.
- No loss in utility compared to traditional approach using shuffling with privacy accounting assuming Poisson subsampling

Future Steps?

- Privacy Accounting for shuffling
  - Only give lower bound
  - Open problem to find tight upper bounds
- Alternative batch samplers