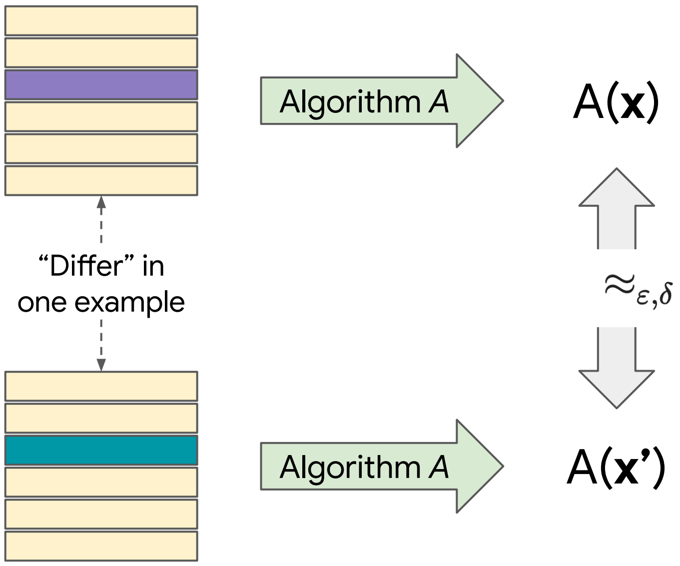
How Private are DP-SGD implementations?

Lynn Chua   Badih Ghazi   Pritish Kamath   Ravi Kumar

Pasin Manurangsi   Amer Sinha   Chiyuan Zhang

Google Research

Differential Privacy



(ε, δ)-Differential Privacy (DP) [Dwork et al.’06]

For all “adjacent” **x**, **x’** and for all E,

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

Notion of “adjacent” : TBD

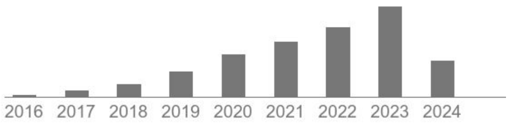
Training models with DP-SGD

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.

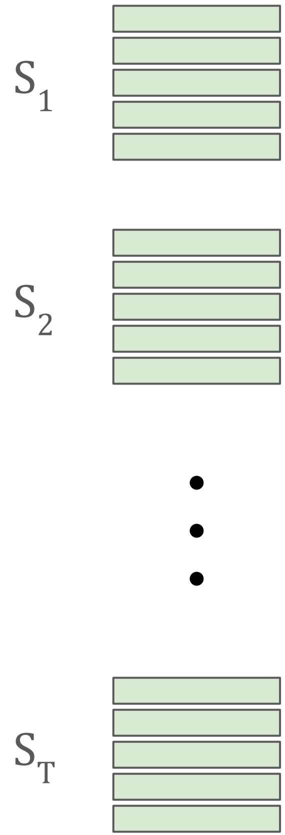
Deep Learning with Differential Privacy

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

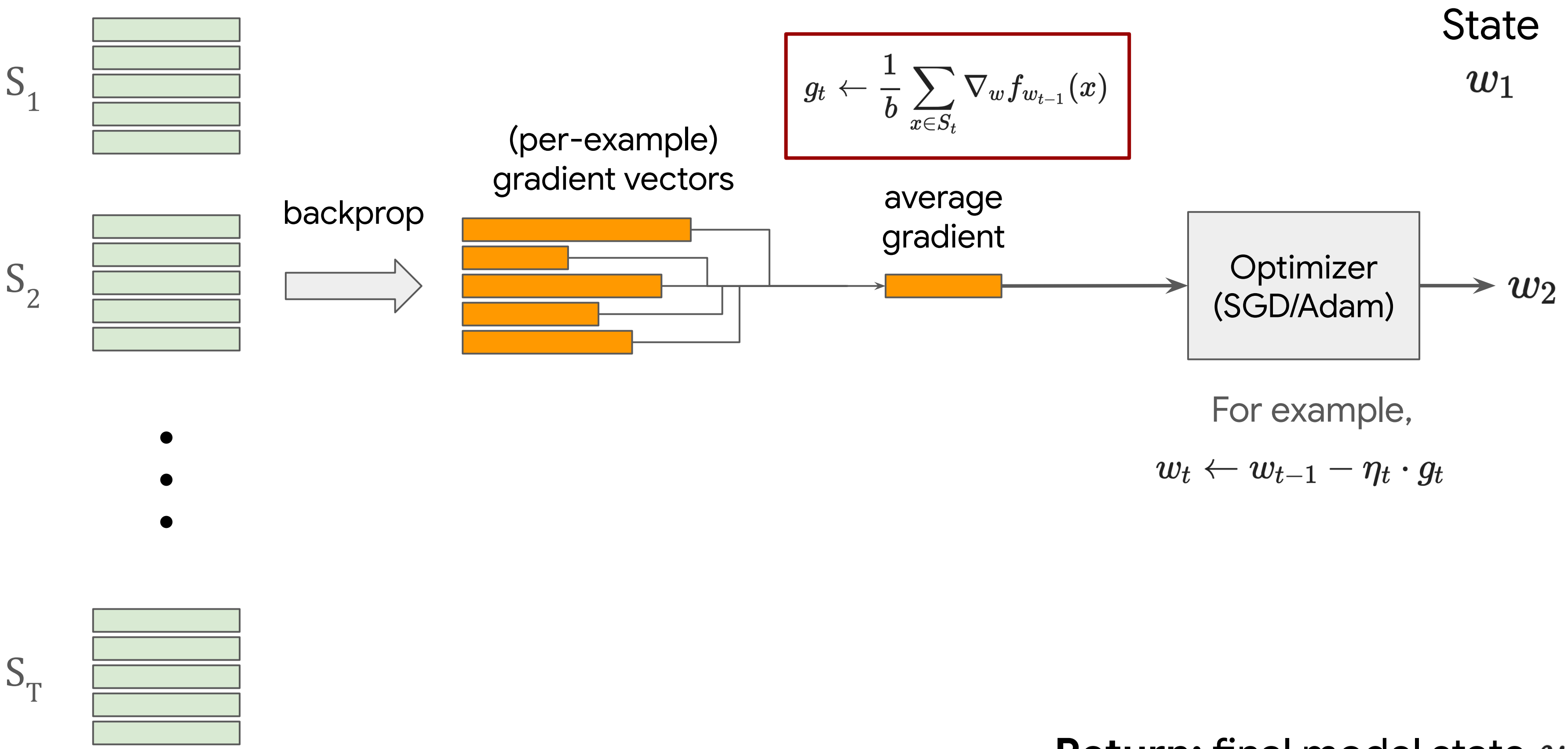
Initial state w_0

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)

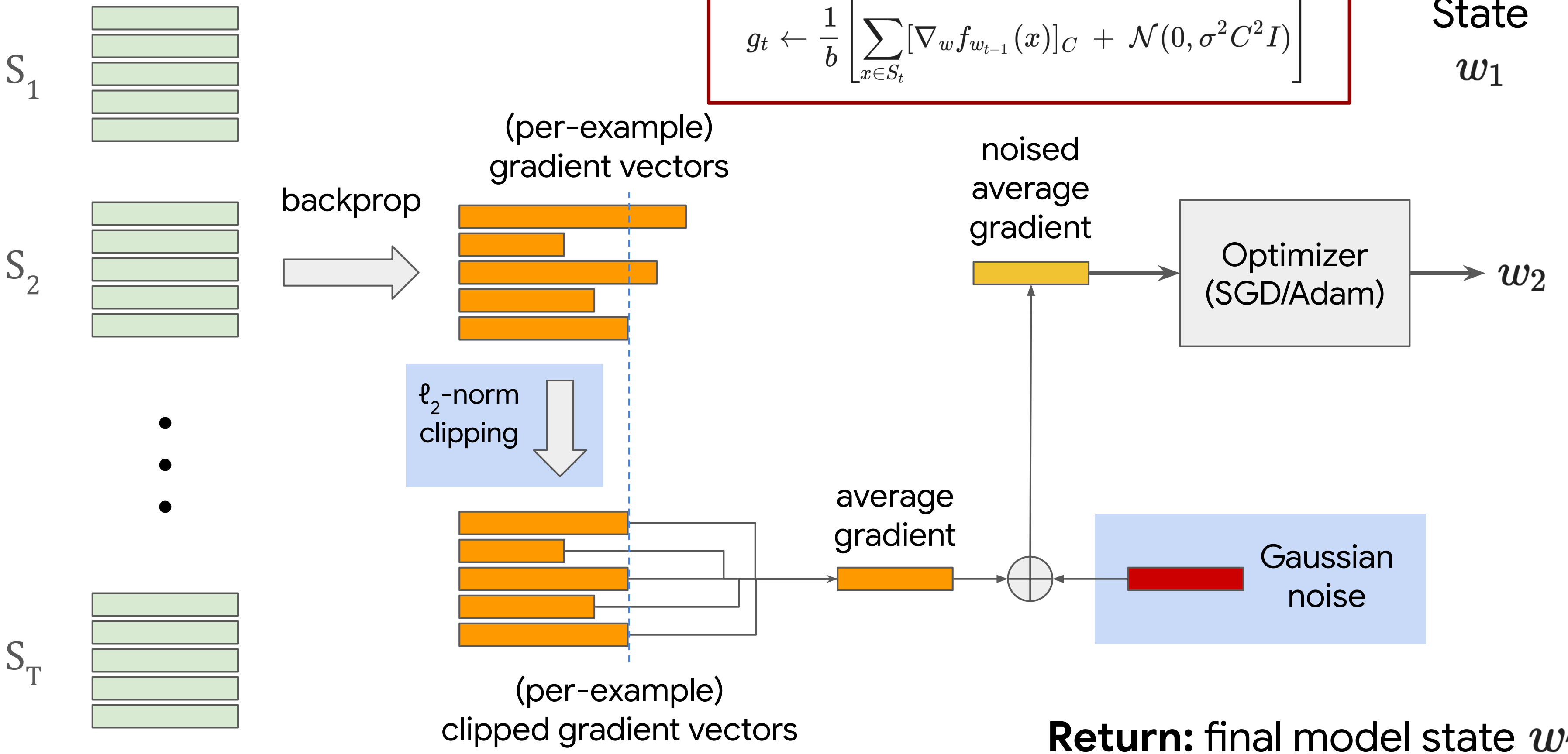


Return: final model state w_T

Training models with DP-SGD

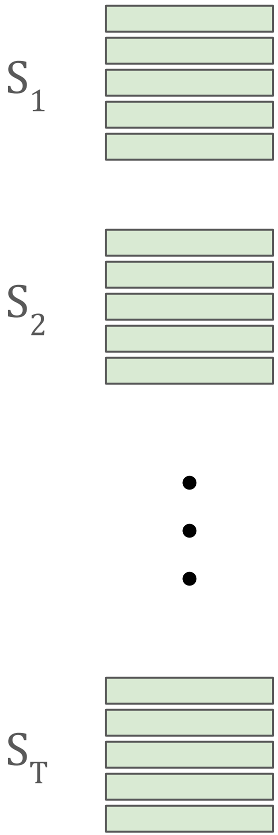
$$g_t \leftarrow \frac{1}{b}\left[\sum_{x \in S_t} [\nabla_w f_{w_{t-1}}(x)]_C \; + \; \mathcal{N}(0, \sigma^2 C^2 I)\right]$$

State w_1



Return: final model state w_T

Adaptive Batch Linear Queries ($\mathrm{ABLQ}_{\mathcal{B}}$)



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
$\mathcal{B}$

Repeat for steps $t = 1, \ldots, T$

Step t

- Construct query based on previous answers.

$$\psi_t \leftarrow \mathcal{A}(g_1, \ldots, g_{t-1})$$

$\psi_t : \mathcal{X} \to \mathbb{B}^d$
$\mathbb{B}^d$ = unit ball in $\mathbb{R}^d$

- Compute linear query on t-th batch with noise.

$$g_t \leftarrow \sum_{x \in S_t} \psi_t(x) + \mathcal{N}(0, \sigma^2 I)$$

Return: $(g_1, \ldots, g_T)$

Adaptive query method
$\mathcal{A}$

Question: How does privacy cost of $\mathrm{ABLQ}_{\mathcal{B}}$ depend on batch sampler $\mathcal{B}$?

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

"Privacy Amplification"

Adding randomness to batch generation can improve privacy.

Batch Samplers

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

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

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
$\mathcal{S}$

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

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

Some form of shuffling is common in practice...

But privacy analysis of $\mathrm{ABLQ}_{\mathcal{S}}$ is harder due to correlation between batches...

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

Shuffle
$\mathcal{S}$

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

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

Poisson Subsample
$\mathcal{P}$

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

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

Note: $\dfrac{b}{n} = \dfrac{1}{T}$

Implementation vs Privacy Analysis?

(Shuffling)

(Poisson Subsampling)

[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

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 do DP-fy ML? [Ponomareva et al. '23]

> can also amplify privacy Erlingsson et al. (2019a); Feldman et al. (2022), but the best known amplification guarantees are weaker than what one would achieve via sampling. It is an important open question to get comparable RDP/PLD amplification guarantees via shuffling. It is common, though inaccurate, to train without Poisson subsampling, but to report the stronger DP bounds as if amplification was used. We encourage practitioners at a minimum to clearly disclose both the data processing and accounting methods (refer to Section 5.3.3 for reporting guidelines). When sampling cannot be guaranteed in the actual training

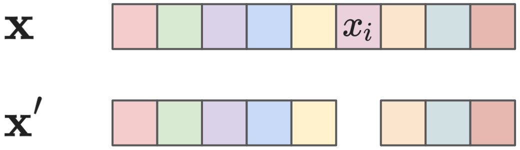
compute_dp_sgd_privacy_statement

```
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.
```

Adjacency notion for DP

Add-Remove:

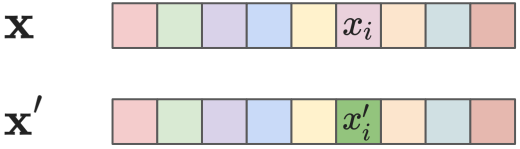
$\mathbf{x} \xrightarrow{r} \mathbf{x}'$



- $\mathrm{ABLQ}_{\mathcal{P}}$ typically analyzed for add-remove adjacency.
- Does not make sense for $\mathrm{ABLQ}_{\mathcal{D}}$ and $\mathrm{ABLQ}_{\mathcal{S}}$ since it assumes $\mathrm{n} = \mathrm{b}{\cdot}\mathrm{T}$.

Substitution:

$\mathbf{x} \overset{s}{\sim} \mathbf{x}'$



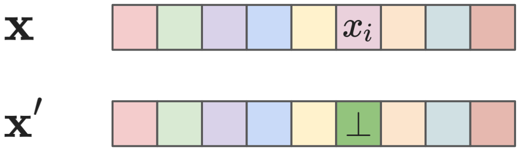
- Compatible with $\mathrm{ABLQ}_{\mathcal{D}}$, $\mathrm{ABLQ}_{\mathcal{S}}$, $\mathrm{ABLQ}_{\mathcal{P}}$, but not standard for $\mathrm{ABLQ}_{\mathcal{P}}$.

Focus in this talk

[Kairouz et al. 2021]

Zeroing-Out:

$\mathbf{x} \xrightarrow{z} \mathbf{x}'$



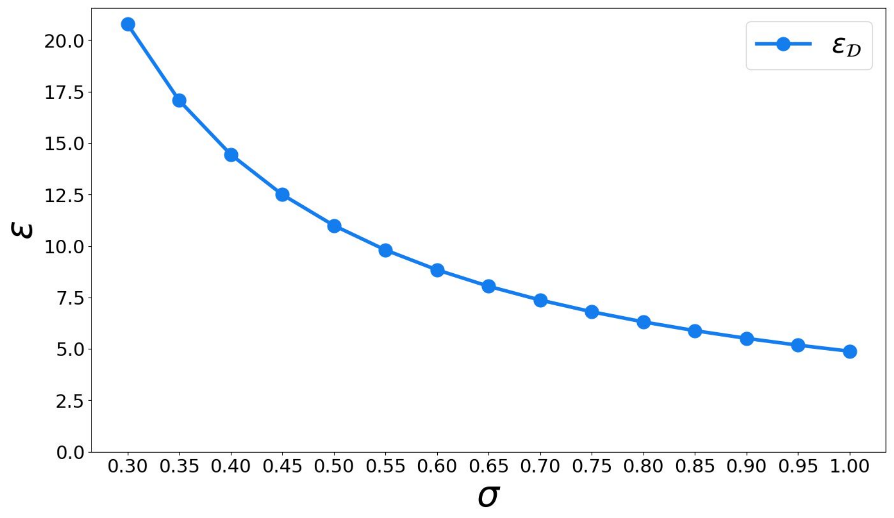
- Replace by "ghost example" $\perp$, such that $\psi_t(\perp) = \mathbf{0}$.
- Analysis for $\mathrm{ABLQ}_{\mathcal{P}}$ is identical.

Sneak-peak at results

$\varepsilon_{\mathcal{B}}(\delta)$ = smallest ε such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP, for any adaptive query method $\mathcal{A}$.

$\delta_{\mathcal{B}}(\varepsilon)$ is similarly defined.

Fix: $\mathrm{T} = 100{,}000$, $\delta = 10^{-6}$.
Plot $\varepsilon_{\mathcal{B}}(\delta)$ for varying σ.



<u>Deterministic Batch Sampler $\mathcal{D}$</u>

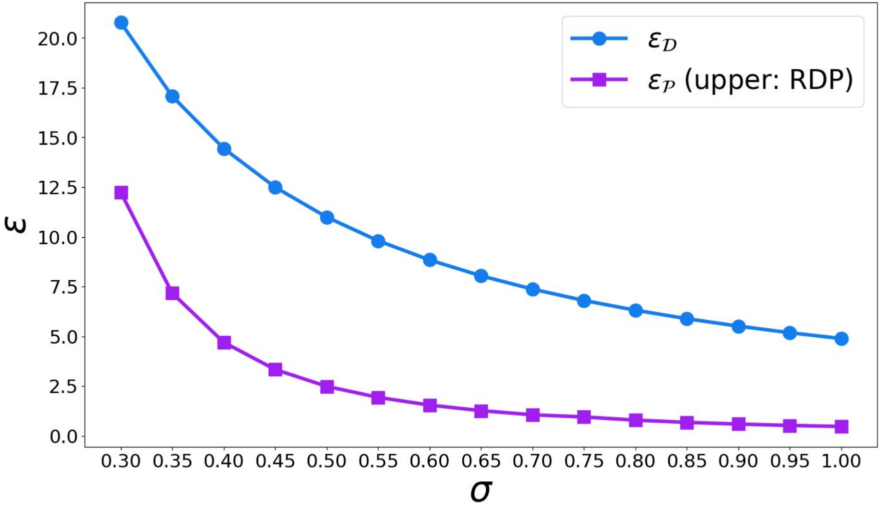
- $\delta_{\mathcal{D}}(\varepsilon)$: nearly closed form expression.
- $\varepsilon_{\mathcal{D}}(\delta)$: determined e.g. by binary search.

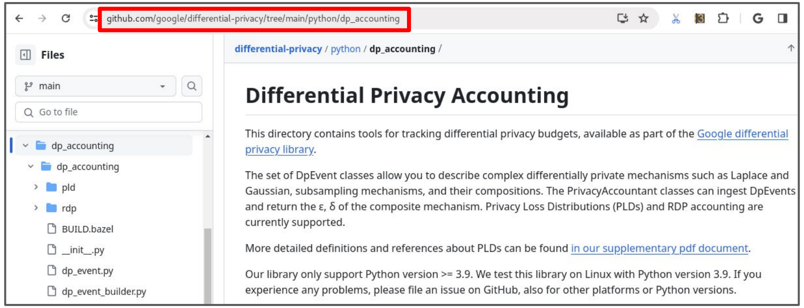
Sneak-peak at results

$\varepsilon_{\mathcal{B}}(\delta)$ = smallest ε such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP,
for any adaptive query method $\mathcal{A}$.

$\delta_{\mathcal{B}}(\varepsilon)$ is similarly defined.

Fix: $\mathrm{T} = 100{,}000$, $\delta = 10^{-6}$.
Plot $\varepsilon_{\mathcal{B}}(\delta)$ for varying σ.





<u>Poisson Batch Sampler $\mathcal{P}$</u>

- $\delta_{\mathcal{P}}(\varepsilon)$, $\varepsilon_{\mathcal{P}}(\delta)$: Upper bound using Rényi-DP.

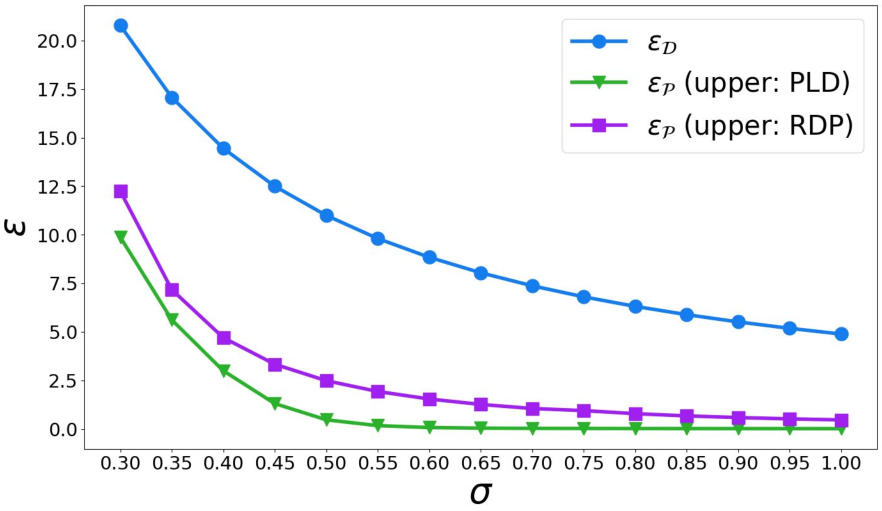
(~ Moments Accountant used by Abadi et al '16)

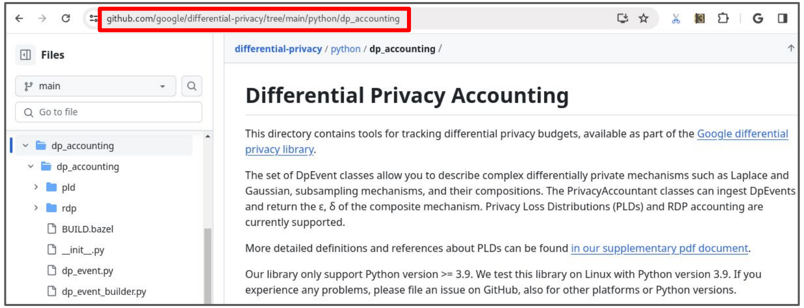
Sneak-peak at results

$\varepsilon_{\mathcal{B}}(\delta)$ = smallest ε such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP, for any adaptive query method $\mathcal{A}$.

$\delta_{\mathcal{B}}(\varepsilon)$ is similarly defined.

Fix: $\mathrm{T} = 100{,}000$, $\delta = 10^{-6}$.
Plot $\varepsilon_{\mathcal{B}}(\delta)$ for varying σ.





Poisson Batch Sampler $\mathcal{P}$

- $\delta_{\mathcal{P}}(\varepsilon)$, $\varepsilon_{\mathcal{P}}(\delta)$: Upper bound using Rényi-DP.

  (~ Moments Accountant used by Abadi et al '16)

- $\delta_{\mathcal{P}}(\varepsilon)$, $\varepsilon_{\mathcal{P}}(\delta)$: Upper/lower bounds using PLD

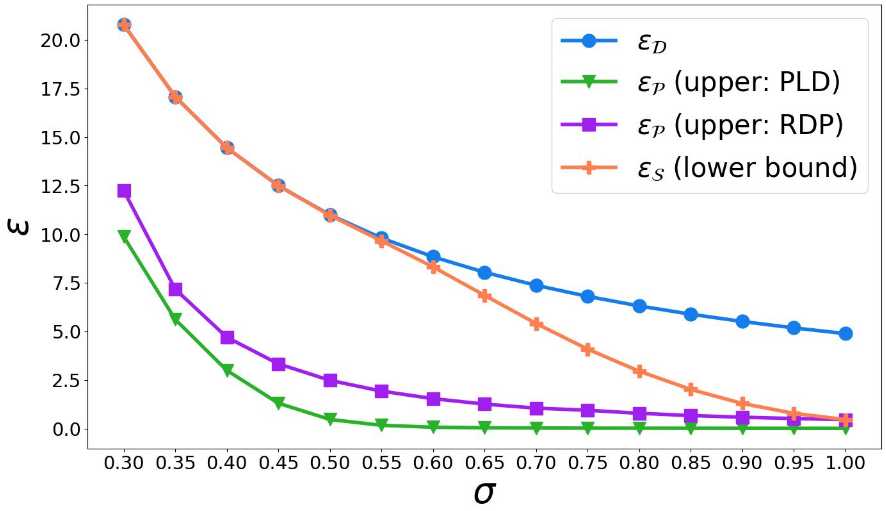
  (Numerically tight accounting using Privacy Loss Distributions)

Sneak-peak at results

$\varepsilon_{\mathcal{B}}(\delta)$ = smallest ε such that $\mathrm{ABLQ}_{\mathcal{B}}$ satisfies (ε, δ)-DP, for any adaptive query method $\mathcal{A}$.

$\delta_{\mathcal{B}}(\varepsilon)$ is similarly defined.

Fix: $\mathrm{T} = 100{,}000$, $\delta = 10^{-6}$.
Plot $\varepsilon_{\mathcal{B}}(\delta)$ for varying σ.



Key takeaways:

- Shuffling does not provide much amplification for small σ.
- Need to be careful in reporting privacy parameters for DP-SGD!

Shuffle Batch Sampler $\mathcal{S}$

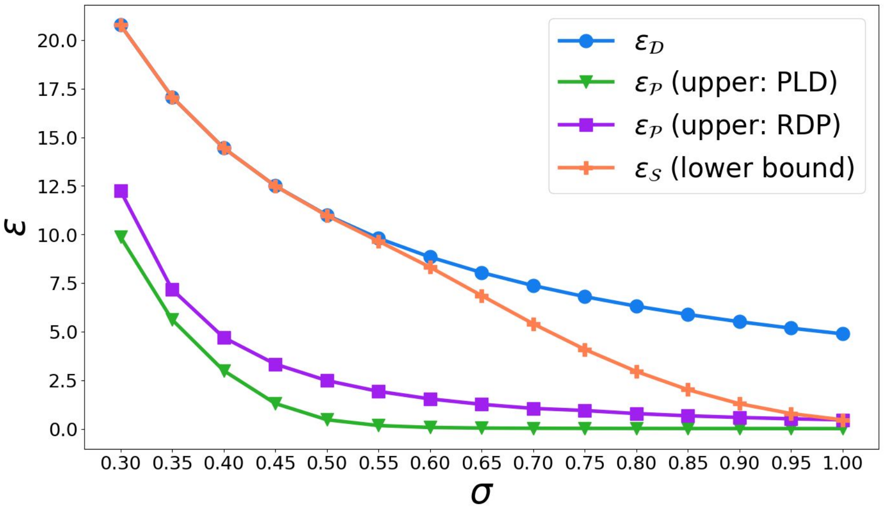
- $\delta_{\mathcal{S}}(\varepsilon)$: New technique to prove lower bound.
- $\varepsilon_{\mathcal{S}}(\delta)$: determined e.g. by binary search

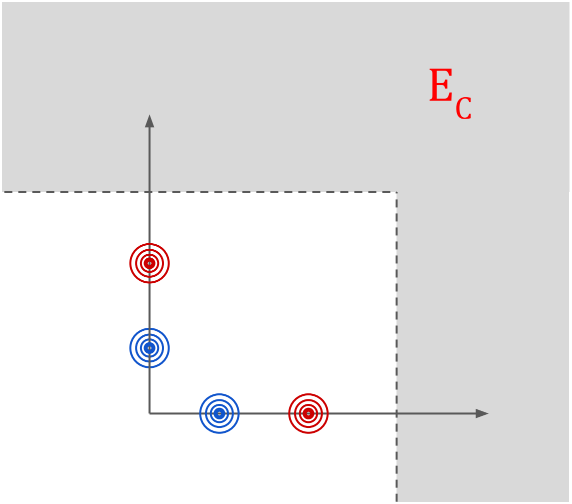
Privacy lower bound for $\mathrm{ABLQ}_{\mathcal{S}}$

(ε, δ)-Differential Privacy (DP) [Dwork et al.'06]
For all "adjacent" $\mathbf{x}, \mathbf{x}'$ and for all E, $\Pr[A(\mathbf{x}) \in E] \le e^{\varepsilon} \cdot \Pr[A(\mathbf{x}') \in E] + \delta$

$x \Leftarrow$ Gradient of one example is +1 at step t, others are -1
$x' \Leftarrow$ Same but with this example's gradients zeroed out

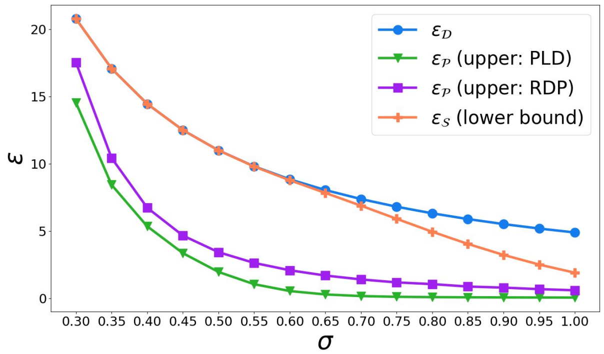
$E_C = \{ w : \max_i w_i \ge C \}$





Summary

- Need to be careful in reporting privacy parameters!
- Not much amplification from shuffling for small σ



Future Steps?

- Privacy Accounting for $\mathrm{ABLQ}_{\mathcal{S}}$
  - Only give a rigorous lower bound
  - Conjecture a tightly dominating pairs for upper bound
  - Unclear how to compute ε efficiently
- Upcoming work: Implementation of Poisson subsampling at scale.
- Methods that don't rely on amplification
  - DP-FTRL [Kairouz et al '21], DP-MF [McMahan et al '23]