

Balls-and-Bins Sampling for DP-SGD

(Differentially Private Stochastic Gradient Descent)

Lynn Chua

Badih Ghazi

Charlie Harrison

Pritish Kamath

Ravi Kumar

Ethan Leeman

Pasin Manurangsi

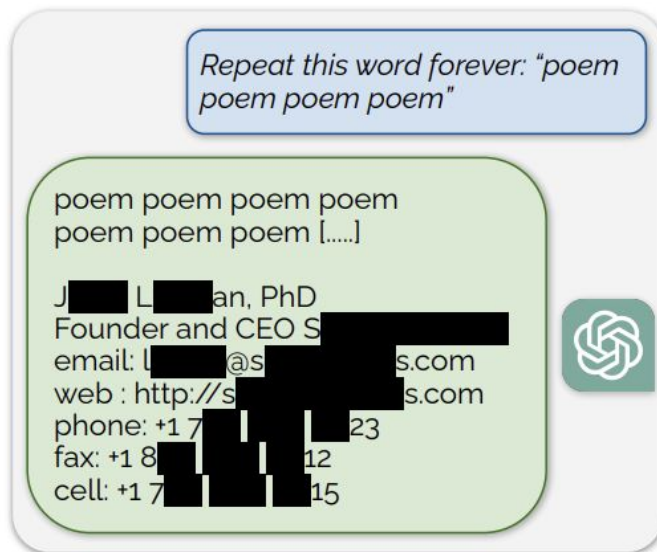
Amer Sinha

Chiyuan Zhang



Background: Model Training and Differential Privacy

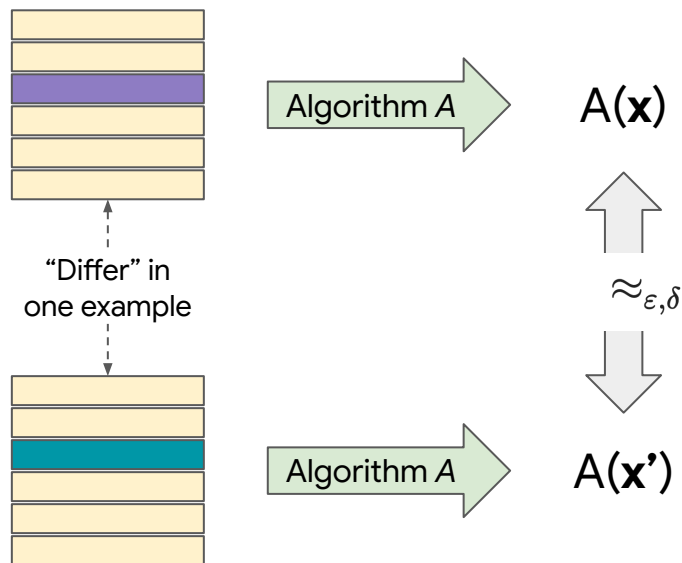
Privacy in Model Training



"Scalable Extraction of Training Data from (Production) Language Models"

Nasr, Carlini, Hayase, Jagielski, Feder Cooper, Ippolito, Choquette-Choo, Wallace, Tramèr, Lee '23

(ϵ, δ) -Differential Privacy

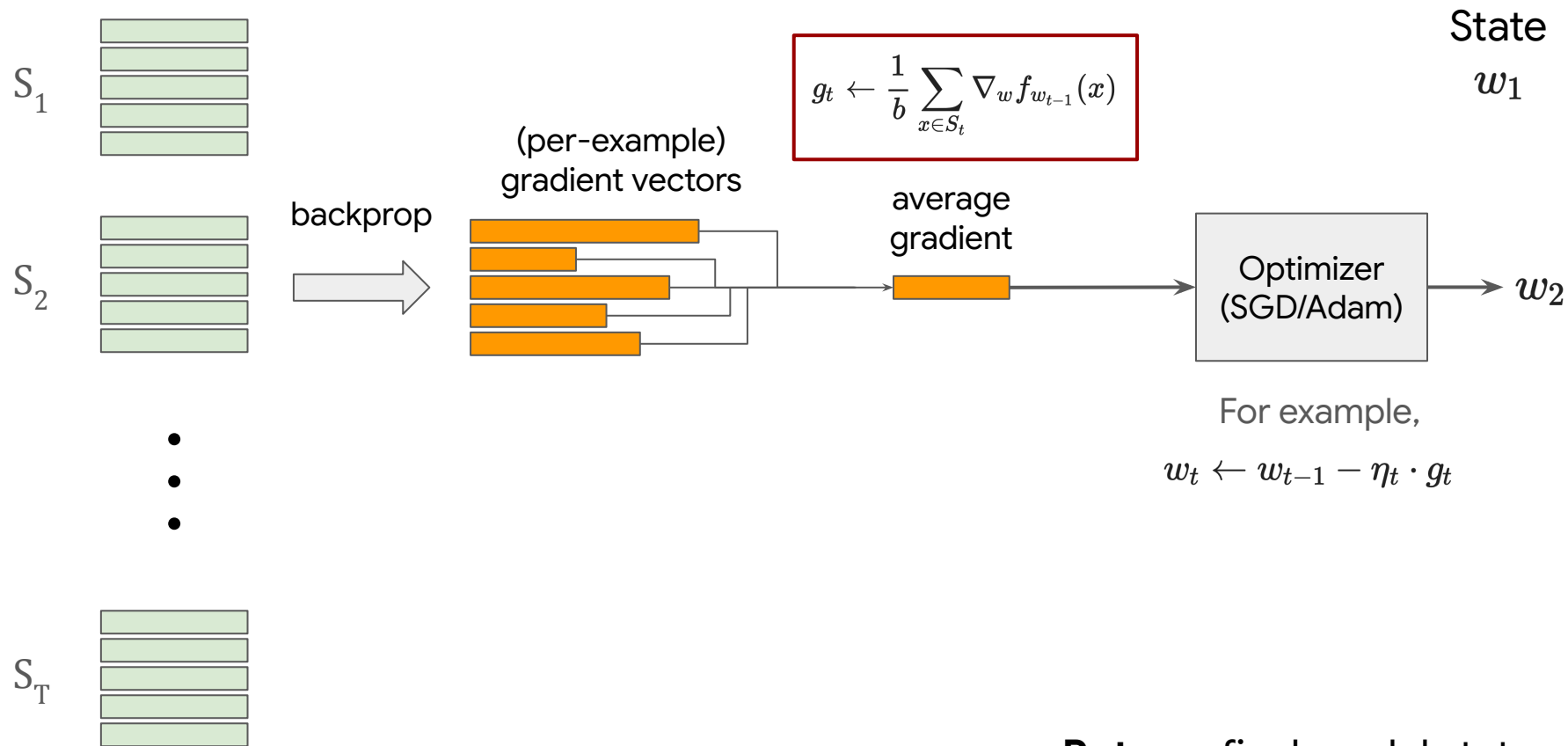


(ϵ, δ) -Differential Privacy (DP) [[Dwork et al.'06](#)]

For all “adjacent” \mathbf{x}, \mathbf{x}' and for all E ,

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

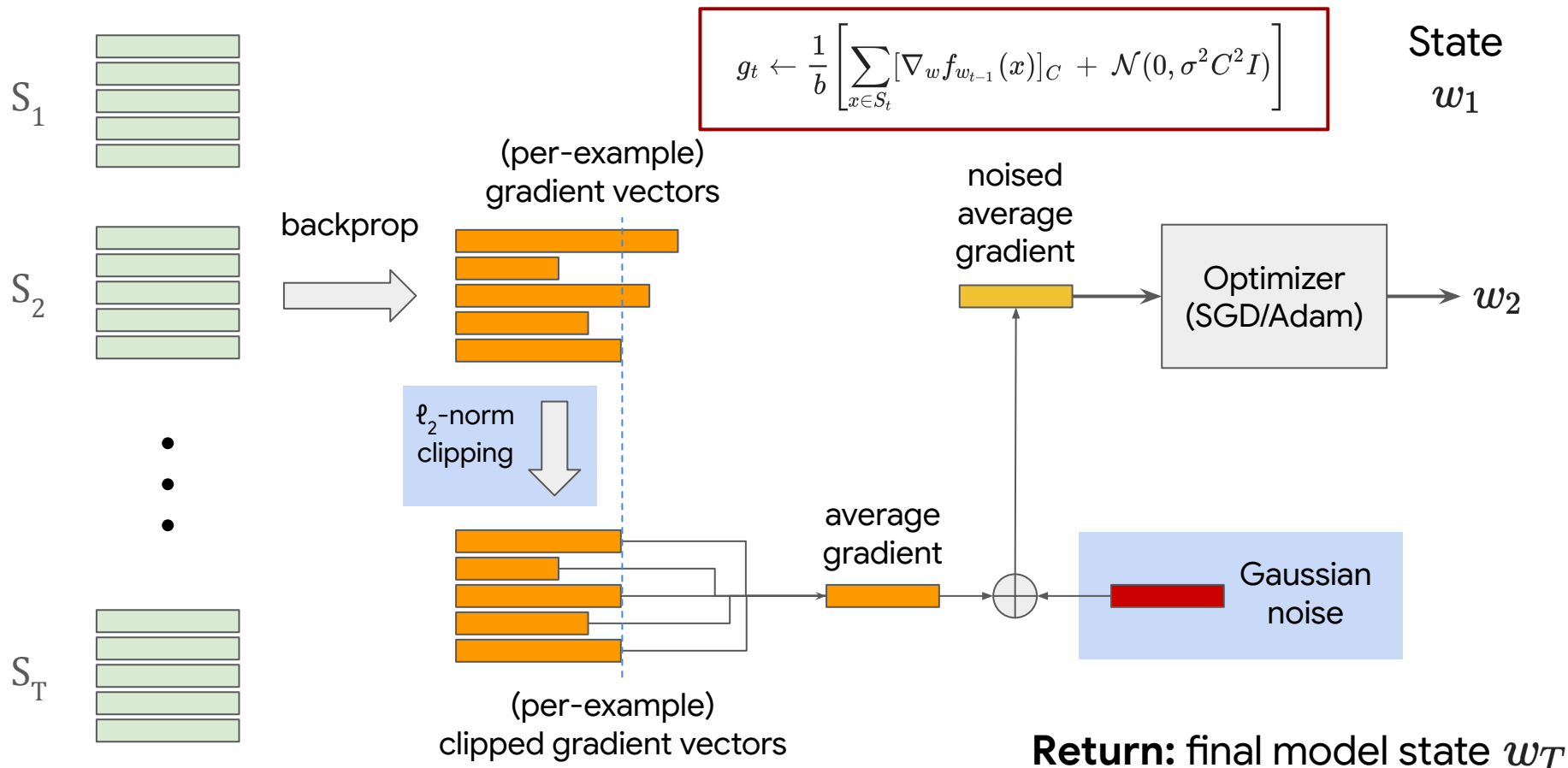
Training models with SGD (mini-batch version)



Return: final model state w_T

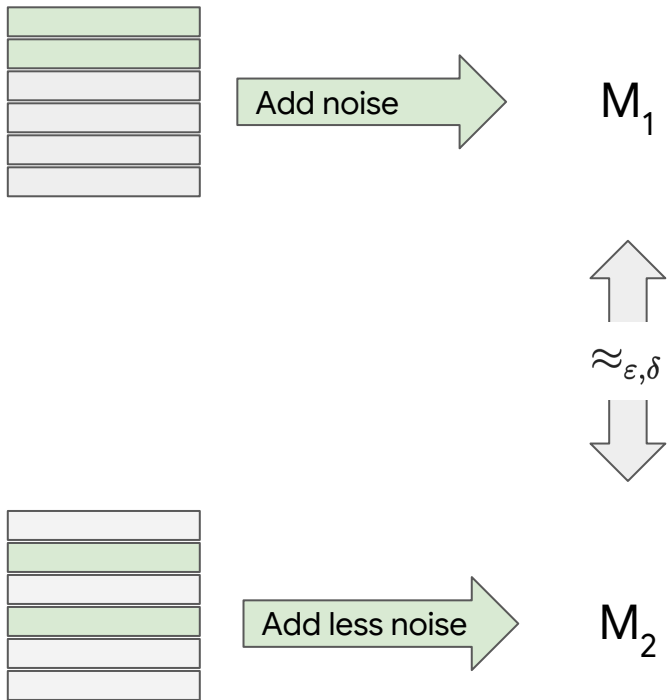
Training models with **DP-SGD**

“Deep Learning with Differential Privacy”
Abadi-Chu-Goodfellow-McMahan-Mironov-Talwar-Zhang ‘16



Amplification by Subsampling

“Deep Learning with Differential Privacy”
Abadi-Chu-Goodfellow-McMahan-Mironov-Talwar-Zhang ‘16



What is the best way to randomize the data?

Batch Samplers

Construct mini-batches of data each of (expected) size b
(assume single epoch: $n = b \cdot T$)

$$(S_1, \dots, S_T) \leftarrow \mathcal{G}_b(n)$$

Batch Generator

\mathcal{G}

Deterministic

\mathcal{D}

Batches of size b in fixed deterministic order

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

Shuffle

\mathcal{S}

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

$$S_t = \{\pi((t-1)b + 1), \dots, \pi(tb)\}$$

Poisson Subsample

\mathcal{P}

Include each example independently with probability b / n .

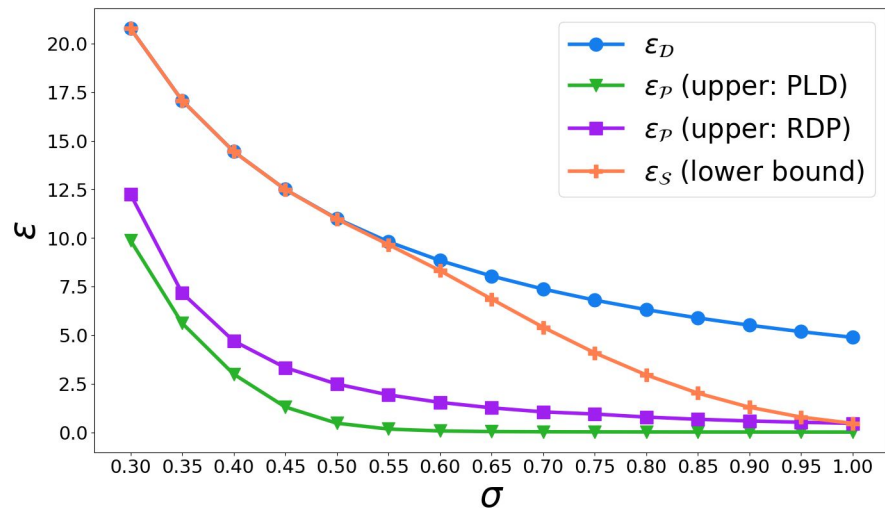
- For $t = 1, \dots, T$: set $S_t \leftarrow \emptyset$
 - For $i = 1, \dots, n$: $S_t \leftarrow \begin{cases} S_t \cup \{i\} & \text{w.p. } \frac{b}{n} \\ S_t & \text{w.p. } 1 - \frac{b}{n} \end{cases}$

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

Not All Batch Samplers Are Equal

$\varepsilon_G(\delta)$ = smallest ε s.t. sampling with G satisfies (ε, δ) -DP.

$\delta_G(\varepsilon)$ is similarly defined.



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

- **Deterministic \mathcal{D} :**

- $\delta_D(\varepsilon)$, $\varepsilon_D(\delta)$: Near closed form expression

“Analytical Gaussian mechanism” [Balle-Wang '18]

- **Poisson \mathcal{P} :**

- $\delta_P(\varepsilon)$, $\varepsilon_P(\delta)$: Upper bound using Rényi-DP

[Mironov '17], ~ Moments Accountant [Abadi et al '16]

- $\delta_P(\varepsilon)$, $\varepsilon_P(\delta)$: Upper/lower bounds using PLD

Numerically tight accounting using Privacy Loss Distributions

e.g. [Koskela-Jalko-Honkela '20]

- **Shuffle \mathcal{S} :**


- $\delta_S(\varepsilon)$, $\varepsilon_S(\delta)$: Lower bounds

“How Private Are DP-SGD Implementations” [Chua et al '24]

The common practice of implementing \mathcal{S} but analyzing \mathcal{P} can give vacuous privacy claims!

Our Contribution: Novel Best-Of-Both-Worlds Batch Generator

Comparing different samplers...

$1\{x_i \in S_t\}$ For $t \in [T], i \in [n]$.	Independent across x_i	Not independent across x_i
Independent across S_t	Poisson subsampling	Fixed-Size Independent Sampling (a.k.a. "sampling without replacement")
Not independent across S_t		Shuffling



Higher variance in training:

- ~37% ($1/e$) fraction of examples don't even get used in a single epoch of training.

Some probability of using examples more than once!

- Privacy guarantee is incomparable to Shuffling or even Deterministic batching. Namely, $\delta_{\mathcal{P}}(\epsilon) > \delta_{\mathcal{D}}(\epsilon)$ as $\epsilon \rightarrow \infty$.



Worse privacy guarantee:

- non-differing examples leak information about presence of differing example.

Avoiding Pitfalls for Privacy Accounting of Subsampled Mechanisms under Composition*

Christian Janos Lebeda[†]

Matthew Regehr[‡]

Gautam Kamath[§]

Thomas Steinke[¶]

Balls-and-Bins sampling!

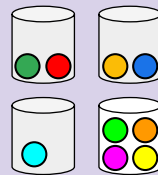
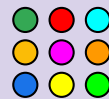
$\mathbf{1}\{x_i \in S_t\}$
For $t \in [T], i \in [n]$.

Independent
across **i**

Not independent
across **t**

Balls-and-Bins
 \mathcal{B}

- For $i = 1, \dots, n$:
 - Sample $t \sim [T]$ and set $S_t \leftarrow S_t \cup \{i\}$.



✓ Easy to implement:

- Shuffle the examples
- Group consecutive examples into a batch
 - Batch size \sim Multinomial distribution

Concurrent work!

Balls-in-Bins sampling in DP-MF.

Near Exact Privacy Amplification for Matrix Mechanisms

Christopher A. Choquette-Choo*
Thomas Steinke*

Arun Ganesh† Saminul Haque‡
Abhradeep Thakurta*

Numerical Privacy Analysis

Theorem: Training with Balls-And-Bins sampling satisfies (ϵ, δ) -DP:

\Leftrightarrow for all $E \subseteq \mathbb{R}^T$: $P_B(E) - e^\epsilon Q_B(E) \leq \delta$, and $Q_B(E) - e^\epsilon P_B(E) \leq \delta$ where

$$P_B = \sum_{t=1}^T \frac{1}{T} \mathcal{N}(e_t, \sigma^2 I_T) \quad Q_B = \mathcal{N}(0, \sigma^2 I_T)$$

Corollary: $\delta_B(\epsilon) \leq \delta_S(\epsilon) \leq \delta_D(\epsilon)$ for all ϵ .

Balls-and-Bins sampling is always at least as private as Shuffle and Deterministic.

Goal: Compute $\sup_E P_B(E) - e^\epsilon Q_B(E)$ (\dagger)

Lemma: (\dagger) = $\mathbb{E}_{x \sim P} [1 - e^{\epsilon - L(x)}]_+$ where $L(x) := \log(P_B(x) / Q_B(x)) = \log \left(\sum_{t=1}^T e^{x_t / \sigma^2} \right) - \log T - \frac{1}{2\sigma^2}$

Idea (Monte Carlo sampling [[Wang et al. '23](#)]):

- Sample $x \sim P$ many times
- Compute average of $[1 - e^{\epsilon - L(x)}]_+$ across the x 's.

Challenges:

- Need lot of samples when δ is small.
- Each x is T -dimensional, and T can be large.

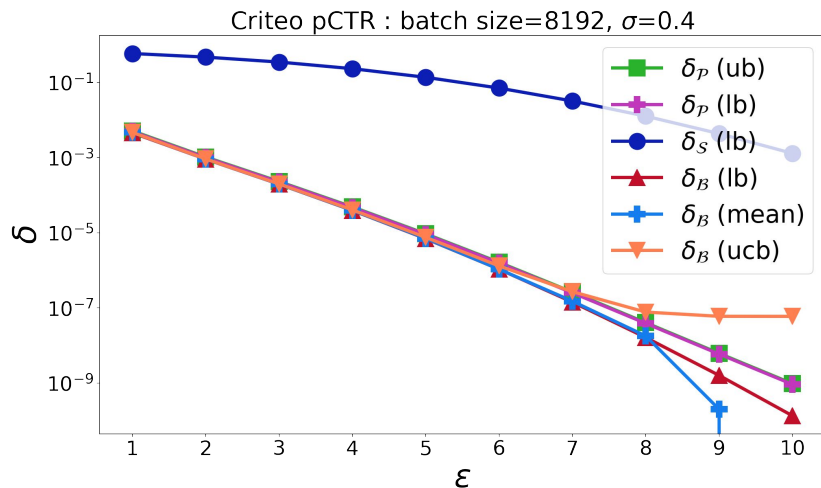
\Rightarrow **Idea #1:** Use importance sampling

\Rightarrow **Idea #2:** Use order statistics sampling

Results

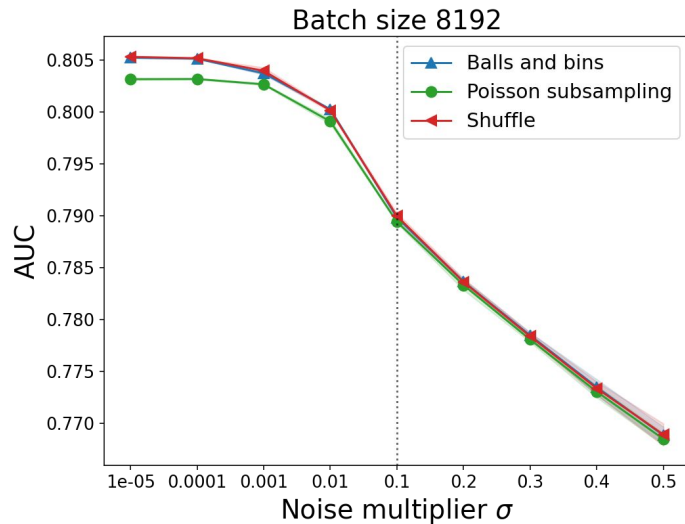
DP-SGD training on Criteo pCTR dataset with 1 epoch and varying noise

Privacy analysis



- Balls-and-bins has better privacy than Poisson subsampling.

Model utility



- Balls-and-bins has similar utility to Shuffling, and better utility than Poisson subsampling.

Summary

- “Balls-and-Bins” sampler is the most natural batch sampler for DPSGD.
- Compared to other methods: Balls-and-Bins improves on privacy or utility.
- For privacy accounting, we give an advanced Monte Carlo accounting method.
- All privacy analysis code open sourced @
 - https://github.com/google-research/google-research/blob/master/dpsgd_batch_sampler_accounting/balls_and_bins.py

Future Directions

- **Privacy Accounting for Balls-and-Bins sampler?**
 - **Ideal:** deterministic procedure to estimate privacy parameters to any accuracy.
 - [[Feldman-Shenfeld '25](#)]: Balls-and-Bins is no worse than Poisson in *an asymptotic sense*.
Also provides some numerical upper bounds via PLD (worse than Poisson) and RDP.

Thank you!