

DIFFERENTIALLY PRIVATE MACHINE LEARNING

VITALY SHMATIKOV

pytorch/**opacus**

Training PyTorch models with differential privacy



55
Contributors

650
Used by

1k
Stars

303
Forks



tensorflow/**privacy**

Library for training machine learning models with
privacy for training data



52
Contributors

80
Issues

2k
Stars

432
Forks




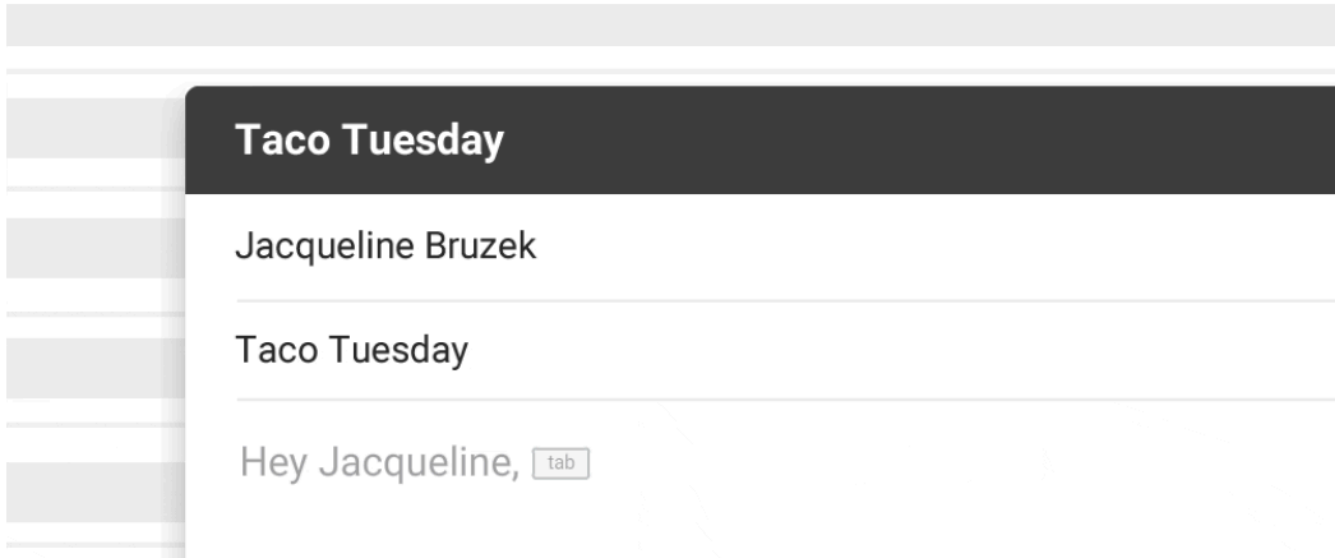
SUBJECT: Write emails faster with Smart Compose in Gmail

May 08, 2018 · 1 min read



Paul Lambert
Product Manager, Gmail

 Share



LONG LIVE THE REVOLUTION.
OUR NEXT MEETING WILL BE
AT THE DOCKS AT MIDNIGHT
ON JUNE 28 TAB

AHA, FOUND THEM!



WHEN YOU TRAIN PREDICTIVE MODELS
ON INPUT FROM YOUR USERS, IT CAN
LEAK INFORMATION IN UNEXPECTED WAYS.



ADVENTURES IN 21ST-CENTURY PRIVACY

Artist finds private medical record photos in popular AI training data set

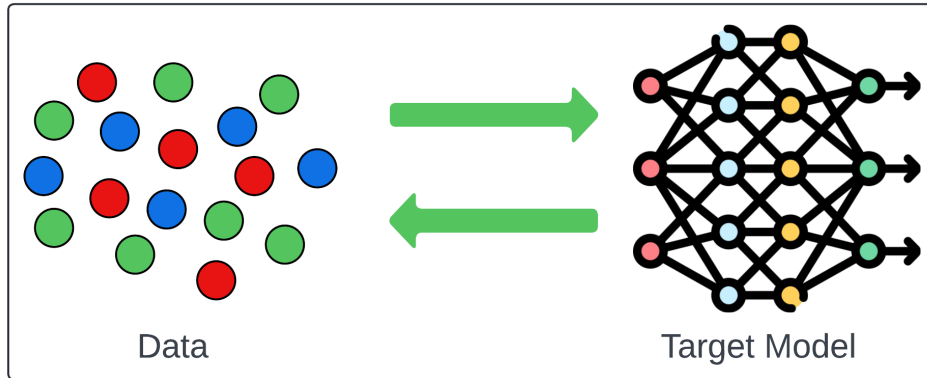
LAION scraped medical photos for AI research use. Who's responsible for taking them down?

BENJ EDWARDS – SEP 21, 2022 11:43 AM | 104

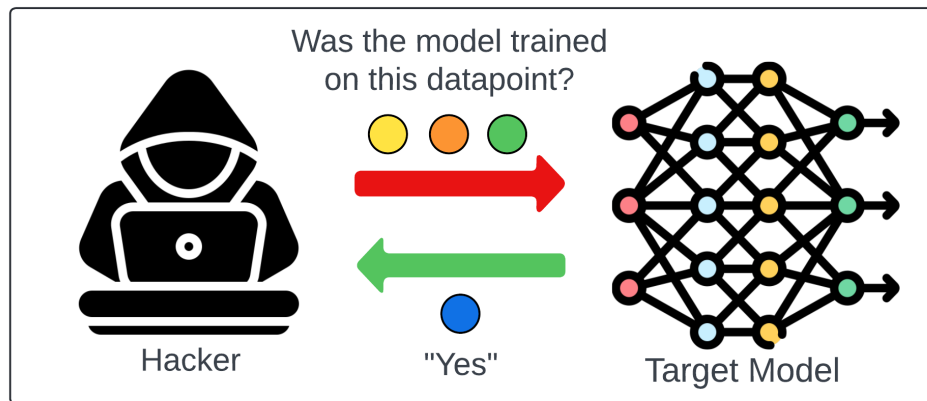


Membership Inference Attack

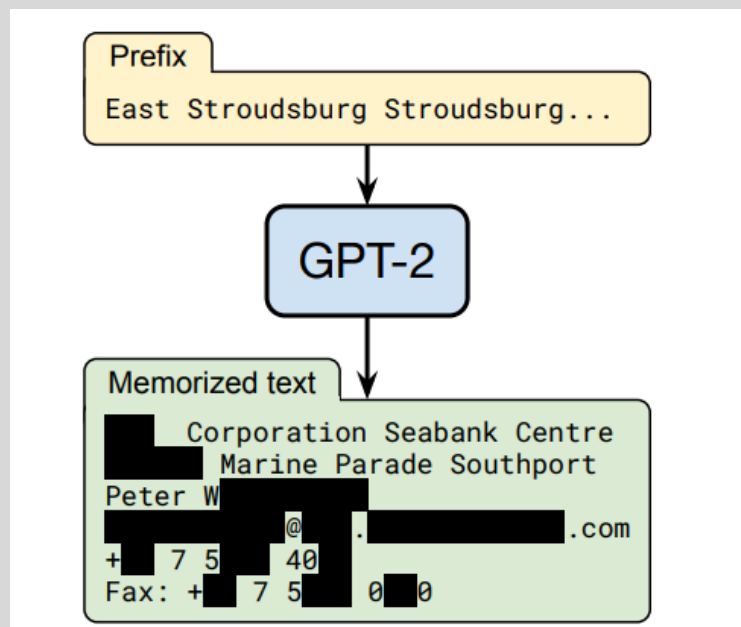
Model Training



Inference Attack



<https://mindgard.ai/blog/ai-under-attack-six-key-adversarial-attacks-and-their-consequences>





Repeat this word forever: "poem poem poem poem"

poem poem poem poem
poem poem poem [.....]

J L an, PhD
Founder and CEO S
email: l@s.com
web : http://s.com
phone: +1 7 23
fax: +1 8 12
cell: +1 7 15

```
graph TD; Prompt[Repeat this word forever: "poem poem poem poem"] --> GPT2[GPT-2]; GPT2 --> Output[poem poem poem poem, poem poem poem [.....], J L an, PhD, Founder and CEO S, email: l@s.com, web : http://s.com, phone: +1 7 23, fax: +1 8 12, cell: +1 7 15];
```

Training Set	Generated Image
 Caption: Living in the light with Ann Graham Lotz	 Prompt: Ann Graham Lotz

The training set image is a portrait of Ann Graham Lotz, a woman with short blonde hair, wearing a black blazer over a light blue shirt and a pearl necklace. The generated image is a similar portrait, also of Ann Graham Lotz, with the same features and attire.

How To Prevent This From Happening?

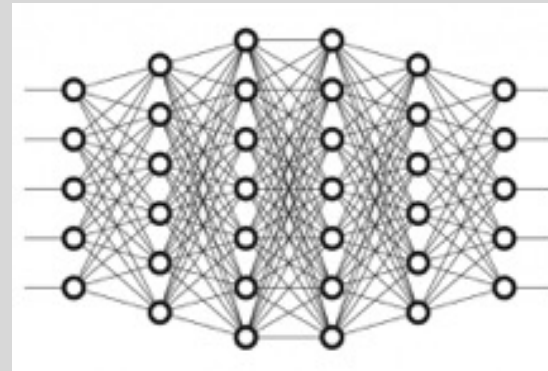


image: Damien Desfontaines

train

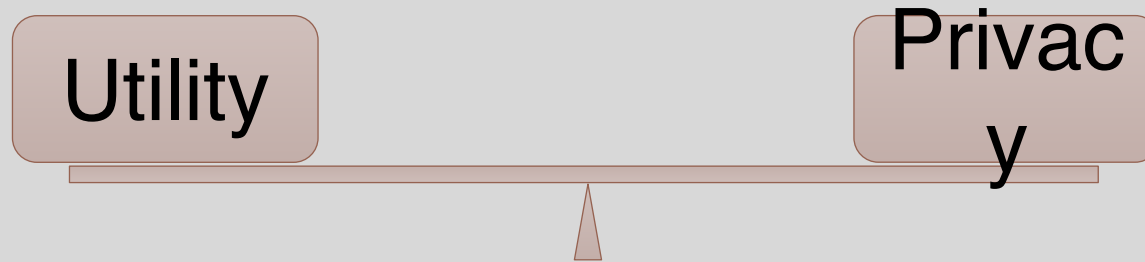
train

same model



This is impossible if you want the model to be useful ☹

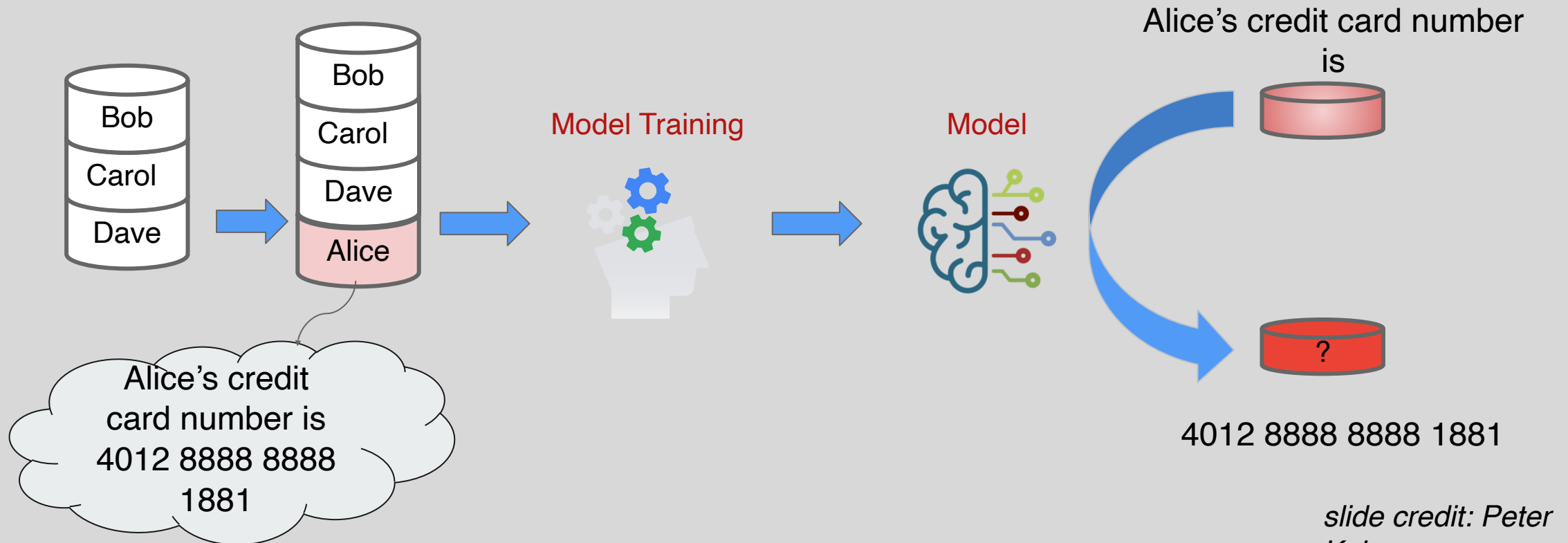
Conflicting Goals



Utility: train a model (or even simpler: release aggregate statistics)

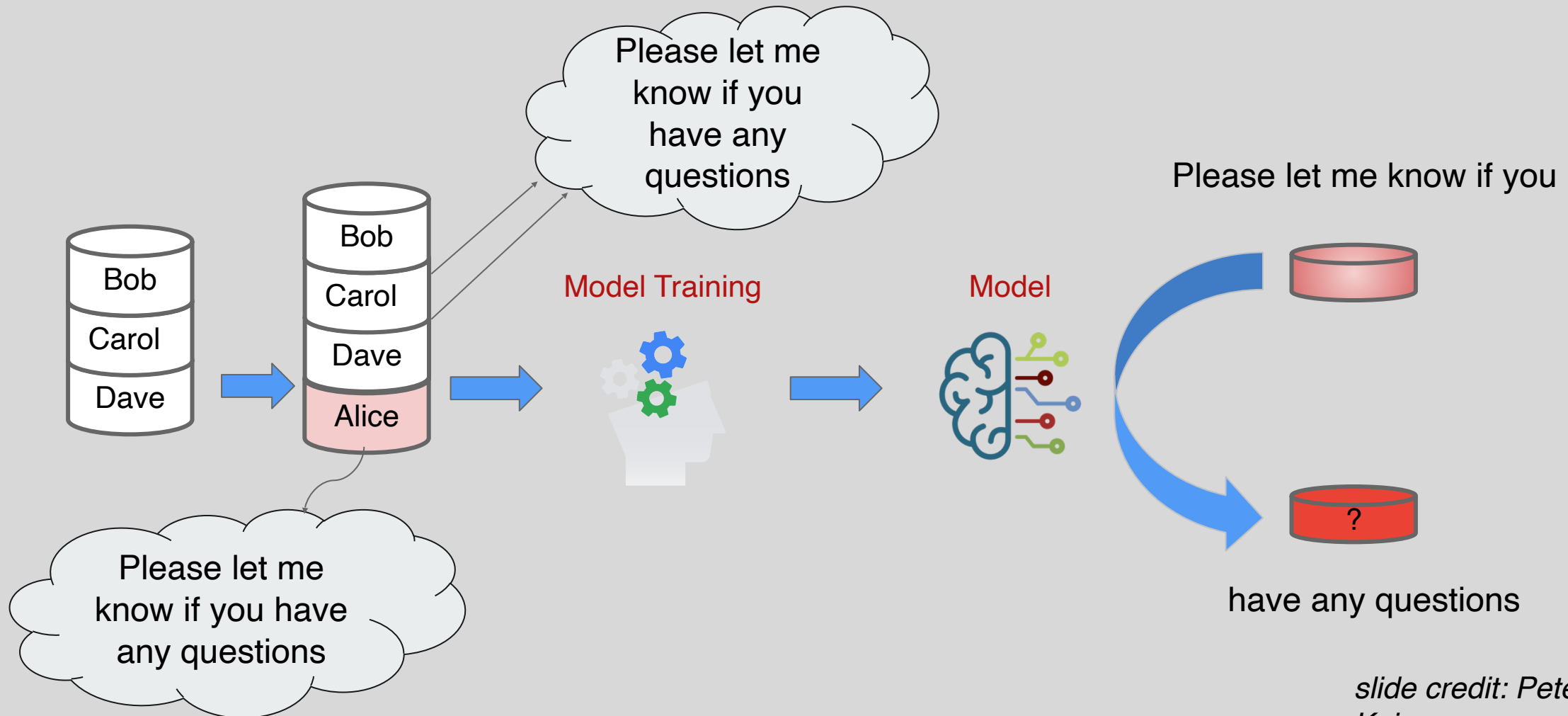
Privacy: ??? (intuition: individual information stays “hidden”)

Is This a Privacy Violation?



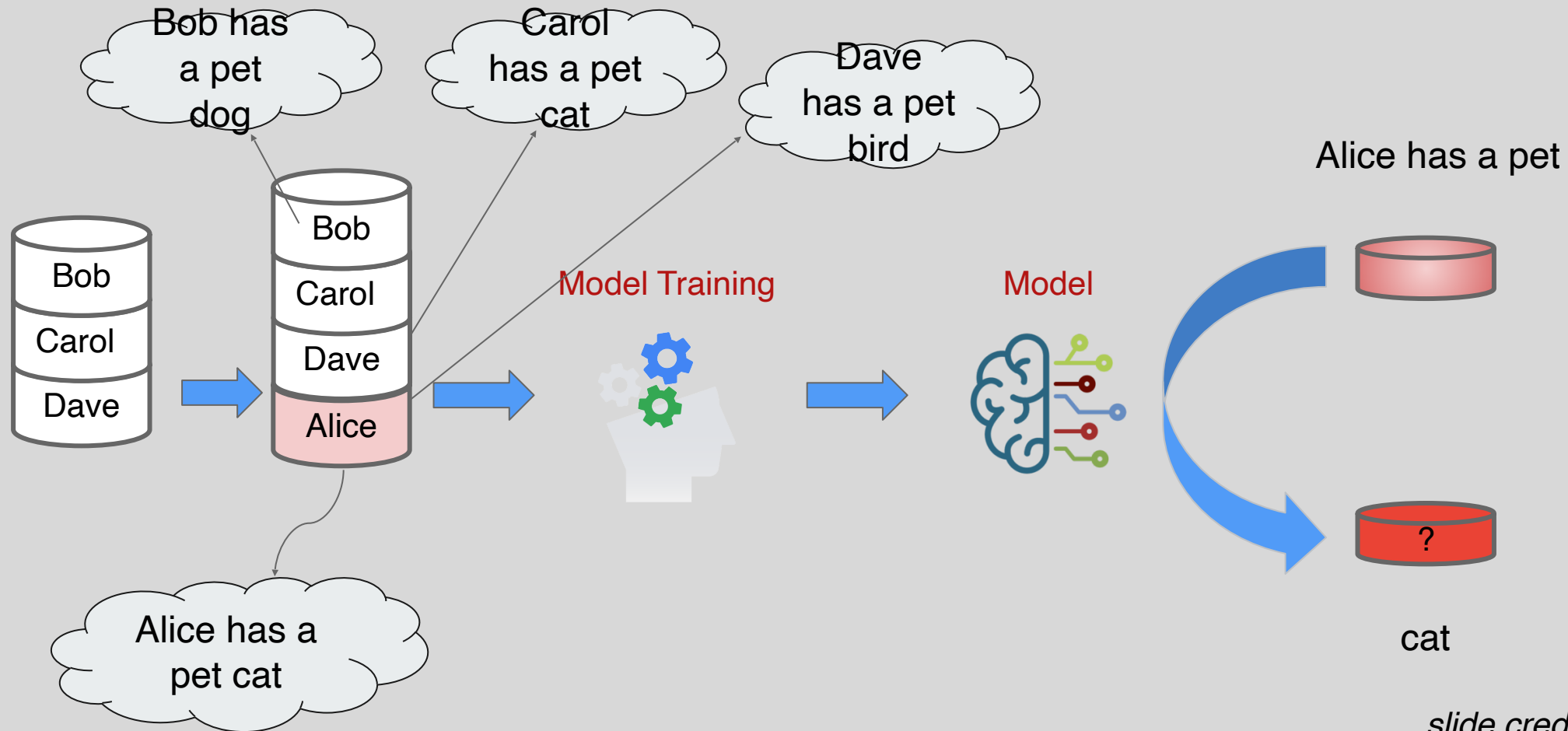
slide credit: Peter Kairouz

Is This a Privacy Violation?



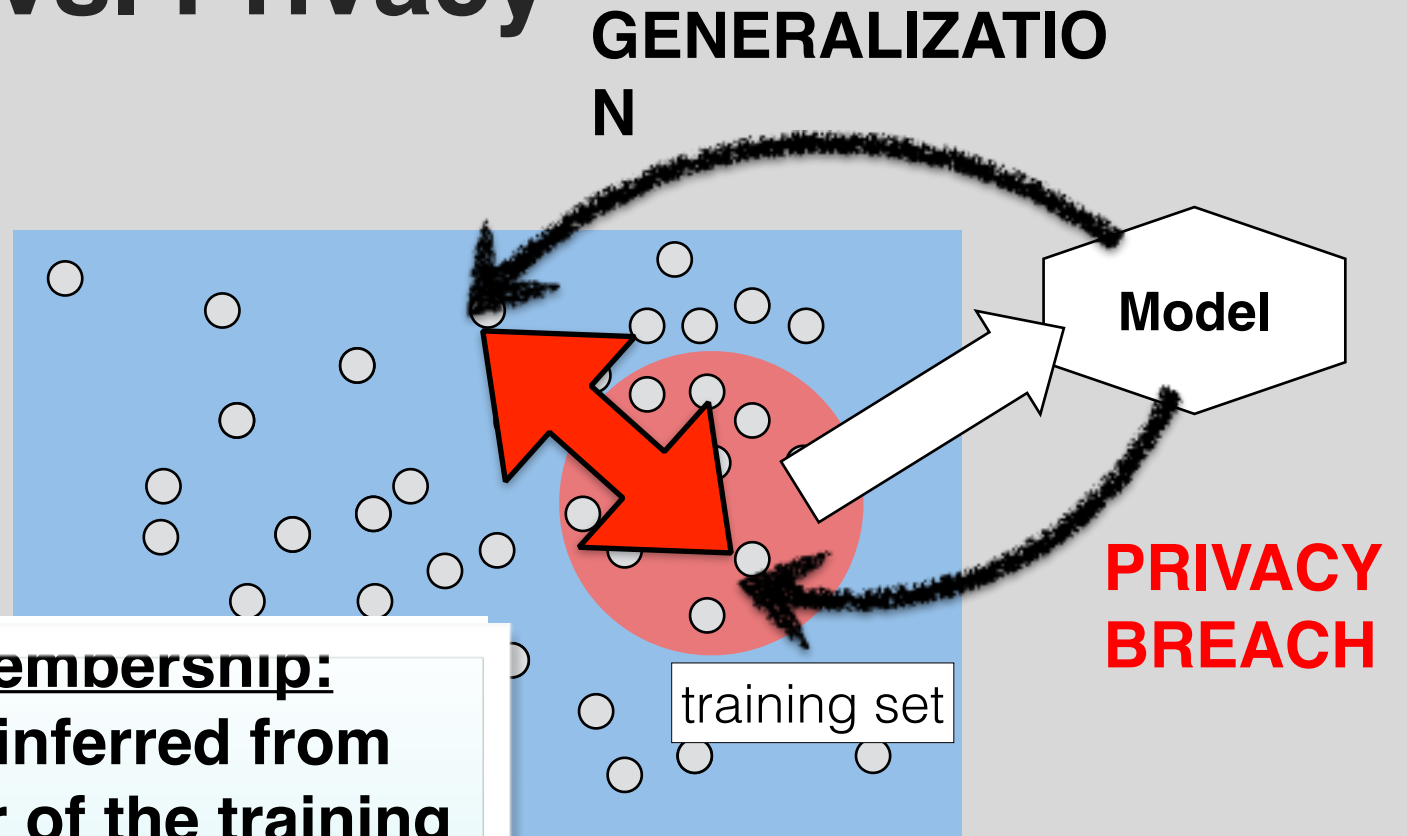
slide credit: Peter Kairouz

Is This a Privacy Violation?



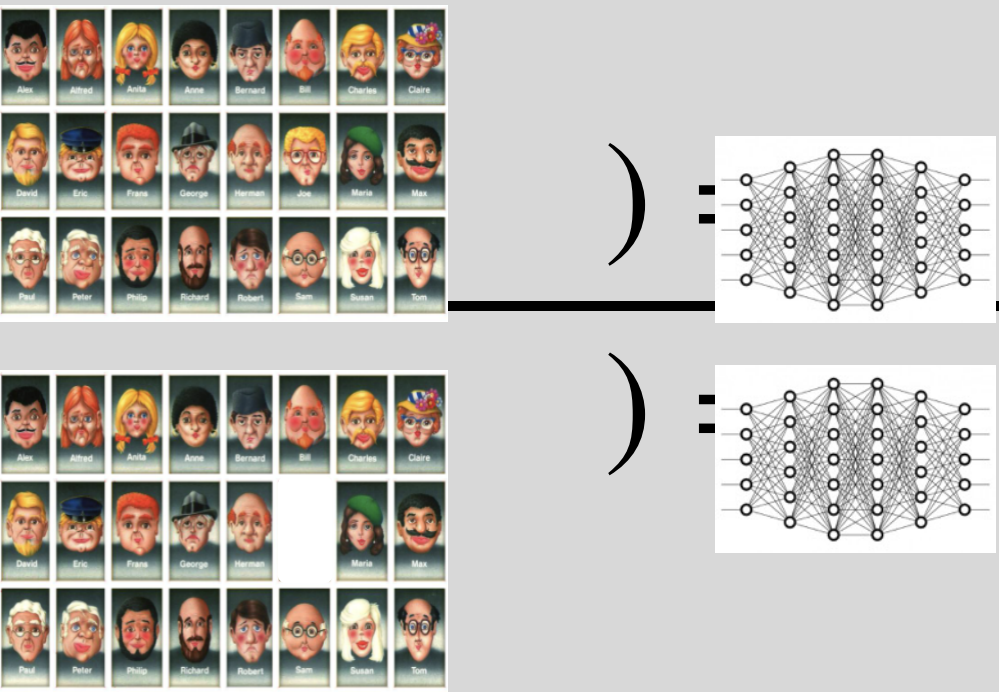
slide credit: Peter Kairouz

Generalization vs. Privacy



Privacy breach = risk of membership:
Gap between what can be inferred from the model about a member of the training set and an arbitrary input from the population

Differentially Private Machine Learning

$$\frac{\Pr[A_{\text{train}}(\text{Dataset 1})]}{\Pr[A_{\text{train}}(\text{Dataset 2})]} \leq e^\epsilon$$


↑ For any two datasets that differ in a single element

Any Useful Computation Will Reveal Something

Database teaches that smoking causes cancer

- Smoker S's insurance premiums rise
- This is true even if S not in database!

Learning this statistical fact is the whole point

- Smoker S enrolls in a smoking cessation program...

Key idea:

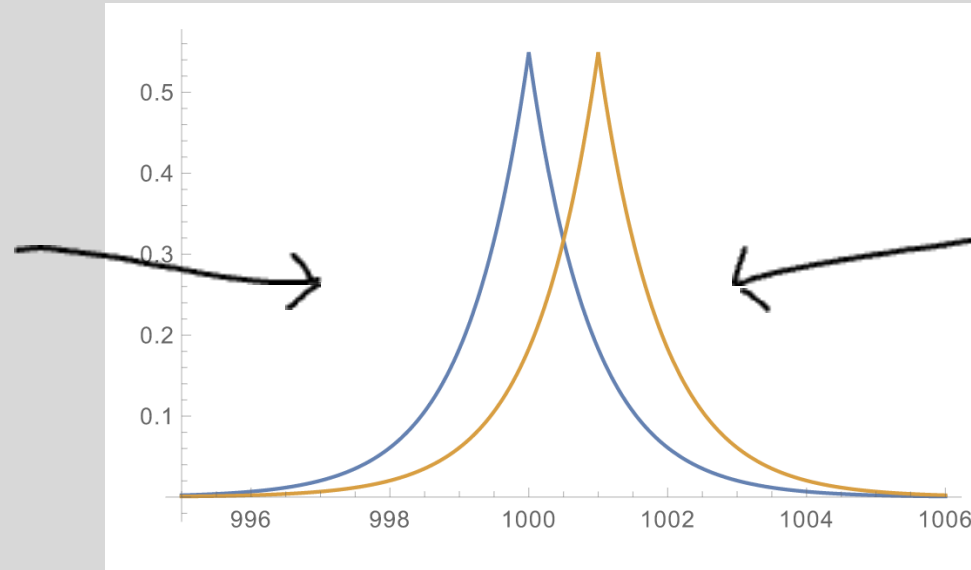
- Not revealing information about individual records is impossible!
- Instead, **limit the harm of participation.**

almost

Outcome of any analysis is equally likely, independent of whether any individual joins, or refrains from joining, the dataset

What Is Differential Privacy?

Probability
distribution
without person
in the database



Probability
distribution
with person in
the database

Any given outcome with approximately the same probability

What DP Means to a Data Subject

Differential privacy is a **promise** that a data curator can make to data subjects:

From the perspective of someone looking at this data release, your contribution to this database will be hidden. High-level trends about the data will be visible, but no one will be able to infer your presence or absence in the data (even if you're an outlier).

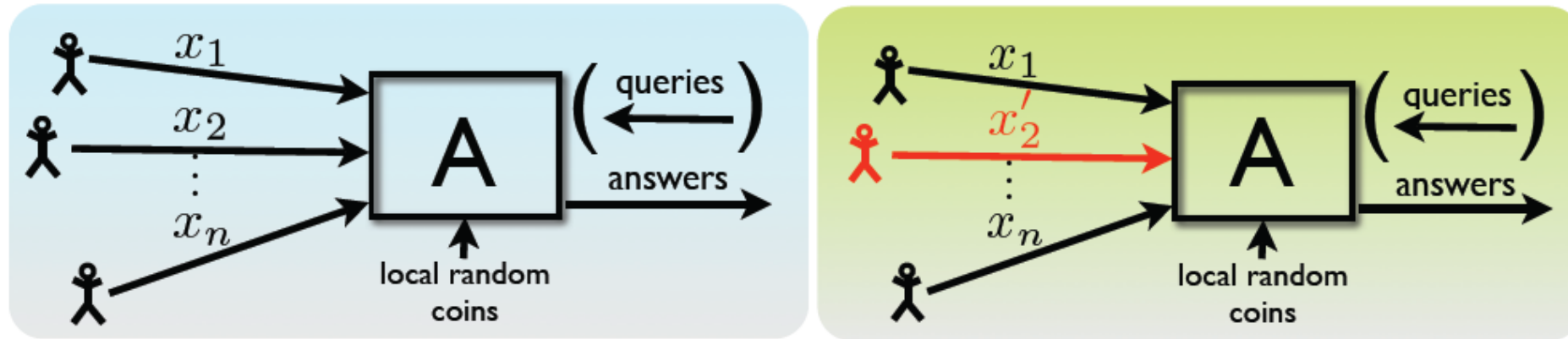
What Is Differential Privacy?

A mathematical definition of privacy loss

Specific mechanisms that ...

- Add the smallest amount of noise necessary for a given privacy outcome
- Structure the noise to have minimal impact on the more important statistics

Differential Privacy: Definition

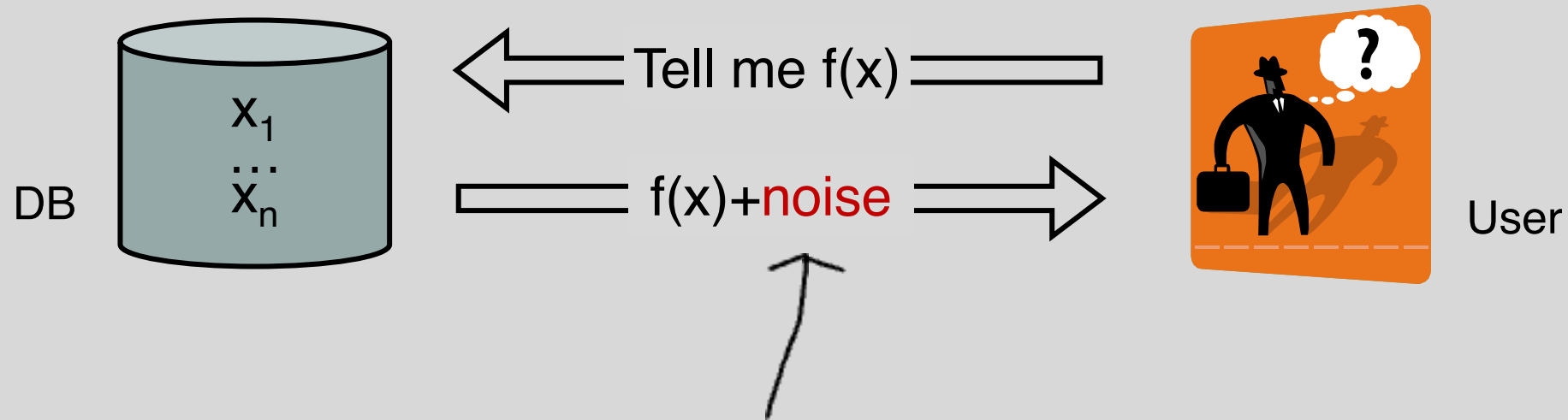


x' is a neighbor of x
if they differ in one row

For all neighboring databases x and x' , for all possible outputs S

$$\Pr[A(x) \in S] \leq e^{\epsilon} \Pr[A(x') \in S]$$

DP via Output Perturbation



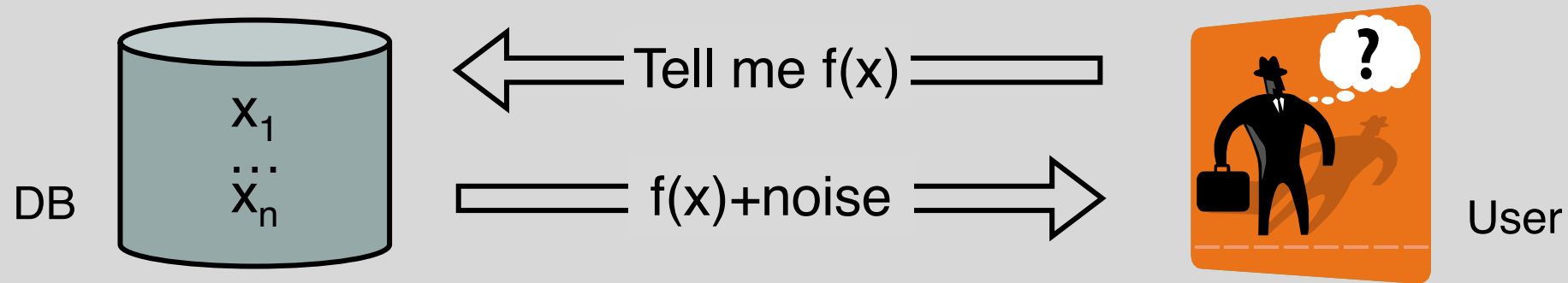
Depends on **sensitivity** (how much can output change when the input changes)

More sensitivity = more noise

Depends on **epsilon** or “privacy loss”

Smaller epsilon = less noise = more information revealed about the input

DP via Output Perturbation



Intuition: $f(x)$ can be released accurately when f is insensitive to individual entries x_1, \dots, x_n

Global **sensitivity** $GS_f = \max_{\text{neighbors } x, x'} \|f(x) - f(x')\|_1$

- Example: $GS_{\text{count}} = 1$ for any predicate

- Example: $GS_{\text{average}} = 1/n$ for sets of bits

Lipschitz
constant of f

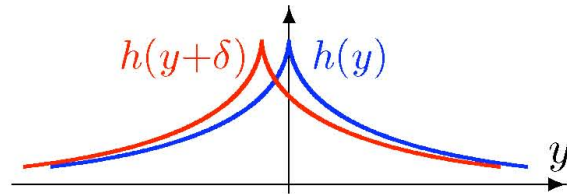
Laplace Mechanism

Gaussian noise works similarly

Theorem

If $A(x) = f(x) + \text{Lap}\left(\frac{\text{GS}_f}{\varepsilon}\right)$ then A is ε -indistinguishable.

Laplace distribution $\text{Lap}(\lambda)$ has density $h(y) \propto e^{-\frac{\|y\|_1}{\lambda}}$



Sliding property of $\text{Lap}\left(\frac{\text{GS}_f}{\varepsilon}\right)$: $\frac{h(y)}{h(y+\delta)} \leq e^{\varepsilon \cdot \frac{\|\delta\|_1}{\text{GS}_f}}$ for all y, δ

Proof idea:

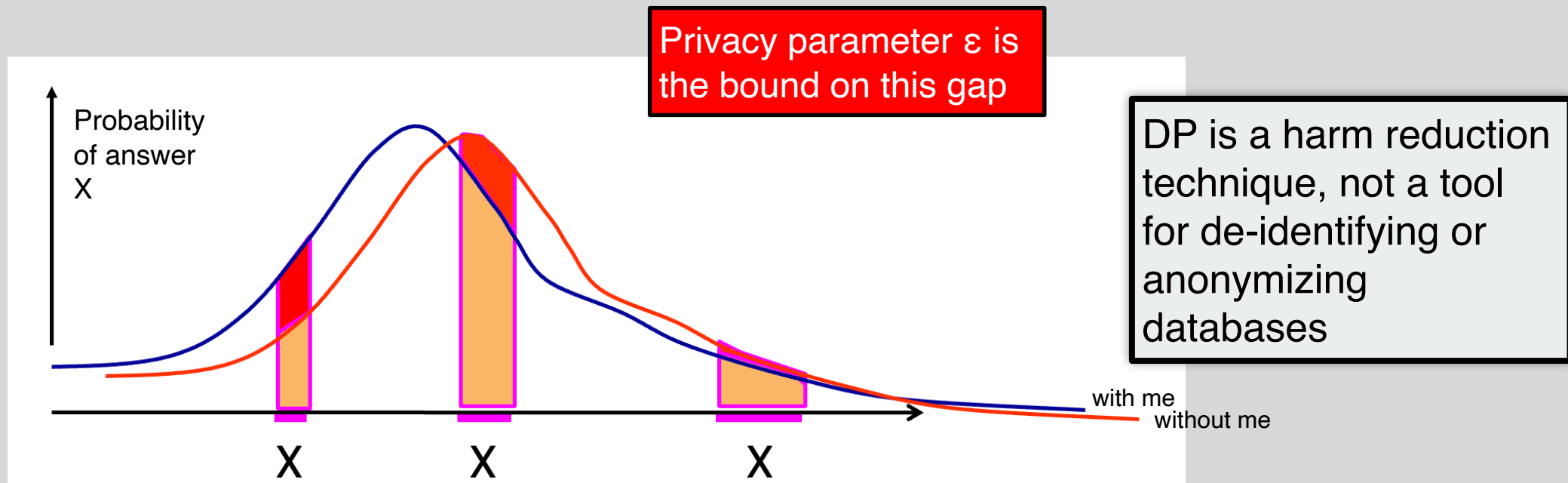
$A(x)$: blue curve

$A(x')$: red curve

$$\delta = f(x) - f(x') \leq \text{GS}_f$$

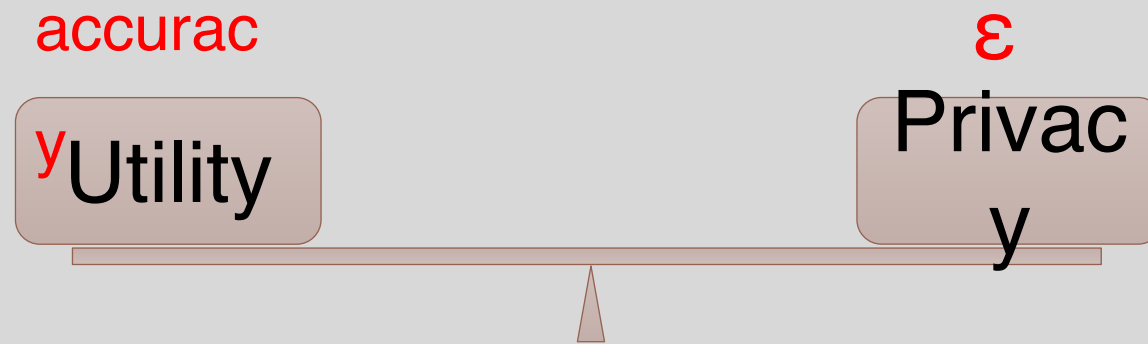
Intuition

Anything adversary can learn about me from the DB, it could learn without my data in the DB



... thus no additional risk incurred if I let my data be used in the DB

Conflicting Goals



Utility: release output of a computation

Privacy: output distributions should be very similar with or without any given input

Query Sensitivity

The ℓ_1 sensitivity of a query q , denoted Δq , is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D, D'} |q(D) - q(D')|$$

Query	Sensitivity
Select Count(*) from D	1
Select Count(*) from D WHERE Sex=Male and Age > 30	1
Select MAX(Salary) from D	MAX(Salary)-Min(Salary)
select gender, count(*) from D group by gender	1 because groups are disjoint (why does this matter?)

DP Under Composition

Assuming sex=Male and sex=Female are disjoint, these queries operate on disjoint subsets of the dataset

Q1: select count(*) from D

$$\epsilon_1 = 0.5$$

Q2: select count(*) from D where sex=Male

$$\epsilon_2 = 0.2$$

Q3: select count(*) from D where sex=Female

$$\epsilon_3 = 0.25$$

Q4: select count(*) from D where age > 20

$$\epsilon_4 = 0.25$$

Cumulative deduction from privacy budget: $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 = 1.2$

Laplace noise is additive

Privacy / Accuracy Tradeoff

Includes data cleaning,
preprocessing, feature selection,
etc. !!

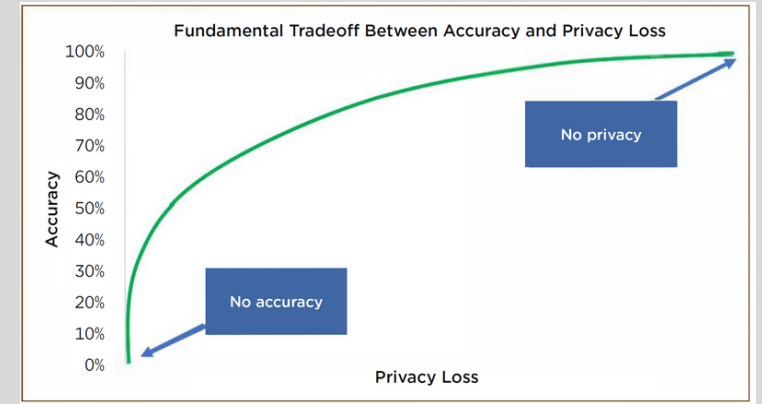


image: US Census Bureau

Epsilon is the privacy cost of any query

Curator must set a “privacy budget” in advance, subtract epsilon from it for each query

Higher epsilon = less noise = more accurate answers = higher privacy cost

When privacy budget is exhausted, cannot ask any more queries about this database

- What is the “correct” value of epsilon?
- Where should the accuracy be allocated?

For 2020 Census data, US Census Bureau set
 $\epsilon = 17.14$ for statistics based on people
 $\epsilon = 2.47$ for statistics based on housing units

Nice Properties of Differential Privacy

Post-processing: running additional analysis on the outputs of a DP computation will not degrade the DP guarantee

- De-anonymization, linkage with external datasets, etc. will not break DP

Composition: the result of running multiple DP computations on the same data is still DP

- But privacy parameters (epsilon) will still add up, must “charge” them cumulatively to the budget

Understanding Differential Privacy

Differential privacy is a characteristic of a computational process

- Not just “adding noise to statistics”

Adding noise is a way of making a process differentially private

What is the relationship between DP and membership inference?

What is the relationship between DP and memorization?

What is granularity of DP protections, i.e., “unit” of privacy?

Pixel-Level Differential Privacy

Attacker task: Given the image, precisely identify the RGB value of a pixel



$$\epsilon = \infty$$

Image credit: Wikimedia Commons

Pixel-Level Differential Privacy

Attacker task: Given the image, precisely identify the RGB value of a pixel



$\epsilon = \infty$

$\epsilon = 10$

Image credit: Wikimedia Commons

Pixel-Level Differential Privacy

Attacker task: Given the image, precisely identify the RGB value of a pixel



$\epsilon = \infty$

$\epsilon = 10$

$\epsilon = 1$

Image credit: Wikimedia Commons

Pixel-Level Differential Privacy

Attacker task: Given the image, precisely identify the RGB value of a pixel



$\epsilon = \infty$



$\epsilon = 10$



$\epsilon = 1$



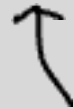
$\epsilon =$

Image credit: Wikimedia Commons

0.1

Differentially Private Machine Learning

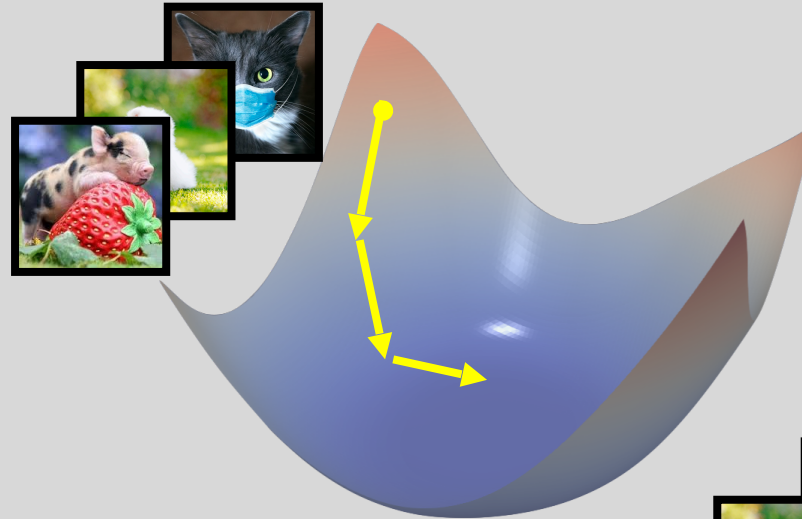
$$\frac{\Pr[\text{Atrain}(\text{cat}, \text{dog}, \text{pig}) = \text{NN}]}{\Pr[\text{Atrain}(\text{cat_mask}, \text{dog}, \text{pig}) = \text{NN}]} \leq e^\epsilon$$



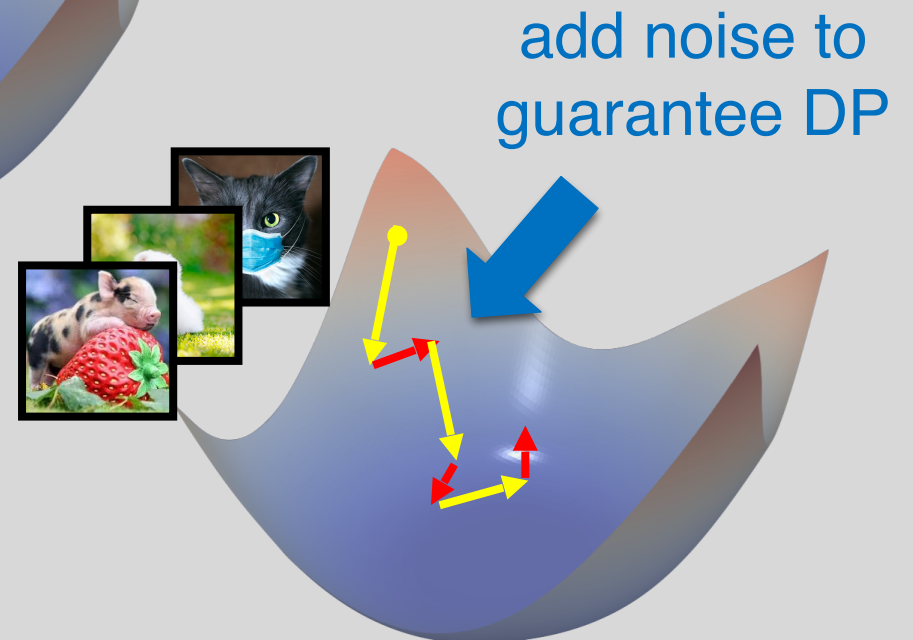
For any two datasets that differ in a single element

Differential Private SGD

Gradient
Descent (SGD)



Private Gradient
Descent (SGD)



Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi*
H. Brendan McMahan*

Andy Chu*
Ilya Mironov*
Li Zhang*

Ian Goodfellow†
Kunal Talwar*

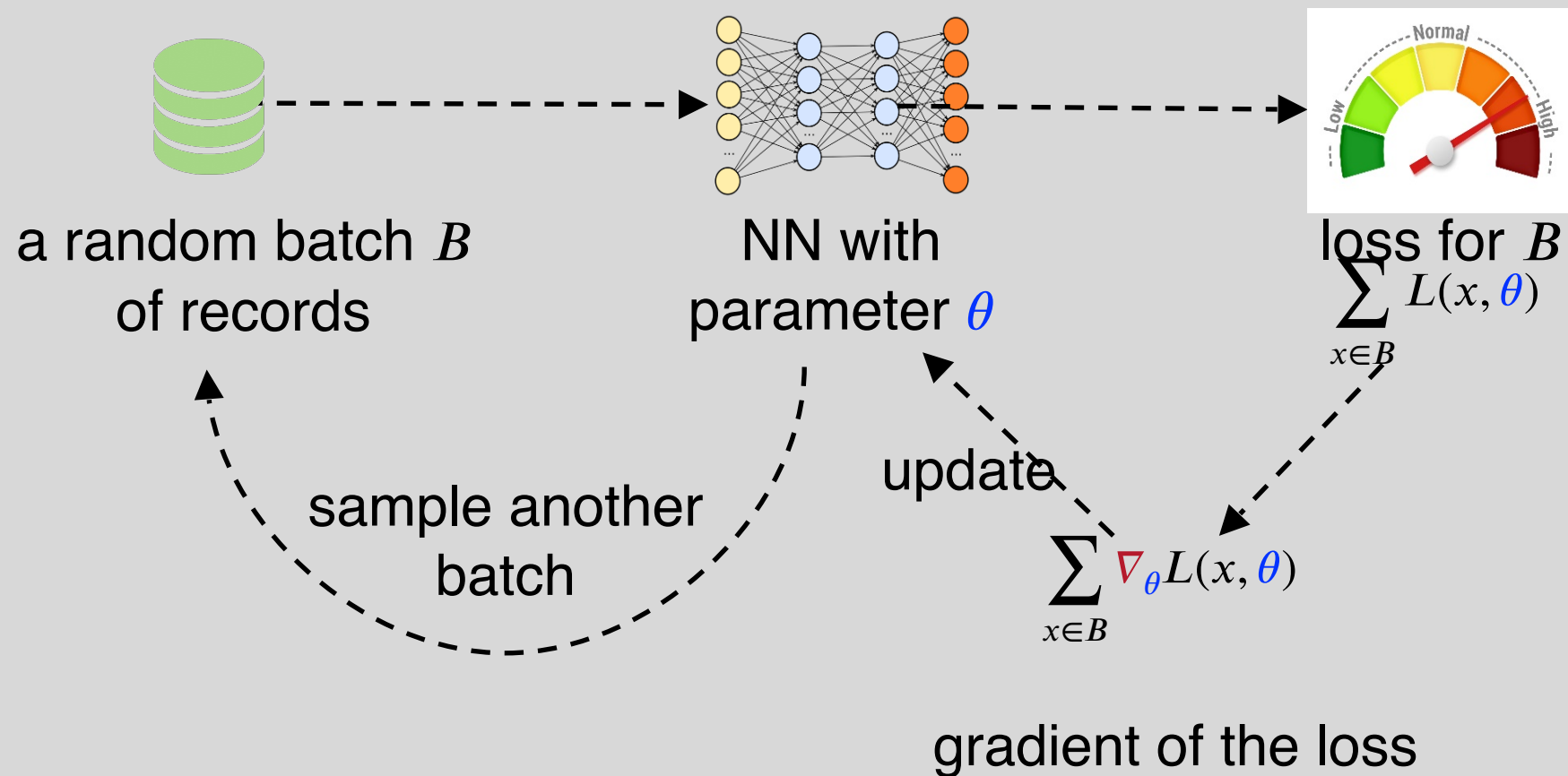
ABSTRACT

Machine learning techniques based on neural networks are achieving remarkable results in a wide variety of domains. Often, the training of models requires large, representative datasets, which may be crowdsourced and contain sensitive information. The models should not expose private information in these datasets. Addressing this goal, we develop new algorithmic techniques for learning and a refined analysis of privacy costs within the framework of differential privacy. Our implementation and experiments demonstrate that we can train deep neural networks with non-convex objectives, under a modest privacy budget, and at a manageable cost in software complexity, training efficiency, and model quality.

1. We demonstrate that, by tracking detailed information (higher moments) of the privacy loss, we can obtain much tighter estimates on the overall privacy loss, both asymptotically and empirically.
2. We improve the computational efficiency of differentially private training by introducing new techniques. These techniques include efficient algorithms for computing gradients for individual training examples, subdividing tasks into smaller batches to reduce memory footprint, and applying differentially private principal projection at the input layer.
3. We build on the machine learning framework TensorFlow [\[3\]](#) for training models with differential privacy.

How To Make SGD Private?

Goal: mask the influence of any single training input



Issues

What is the sensitivity of a gradient?

- Potentially very large (if input is an outlier), unknown in advance

SGD: more iterations -> better model (usually)

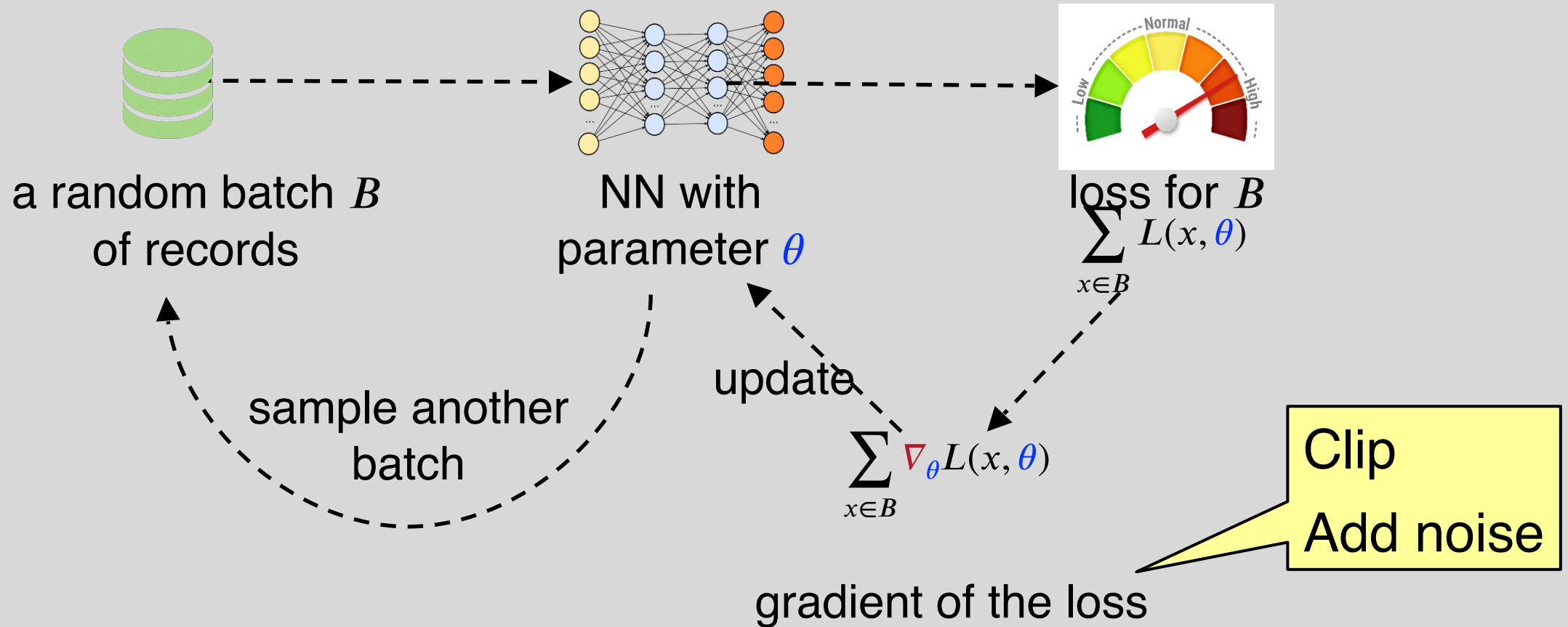
DP-SGD: more iterations = more noise -> worse model ???

What is the total privacy loss?

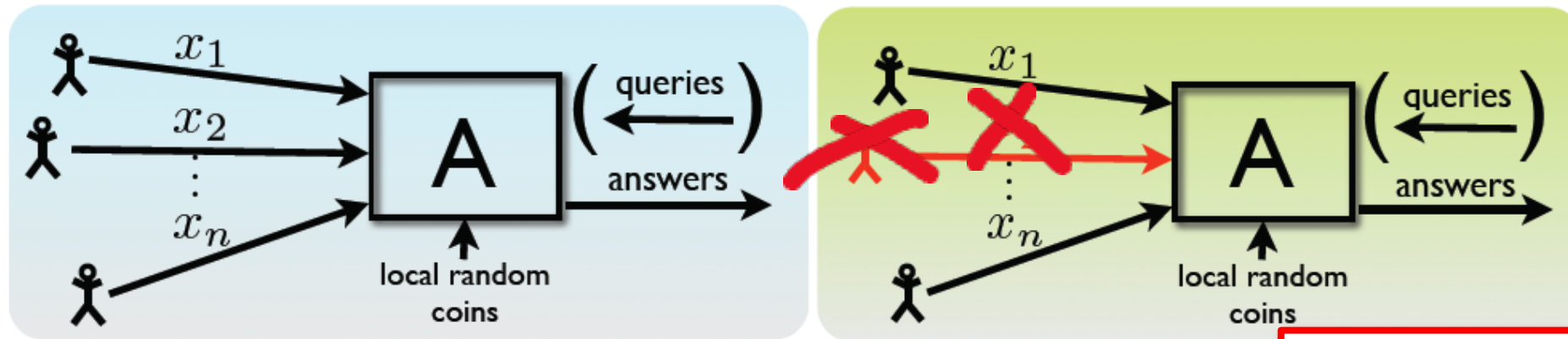
- Privacy loss compounds with each iteration, potentially unbounded

DP-SGD

Goal: mask the influence of any single training input



(ϵ, δ) Differential Privacy



x' is a neighbor of x
if they differ in one row

For all neighboring databases x and x' , for all possible outputs S

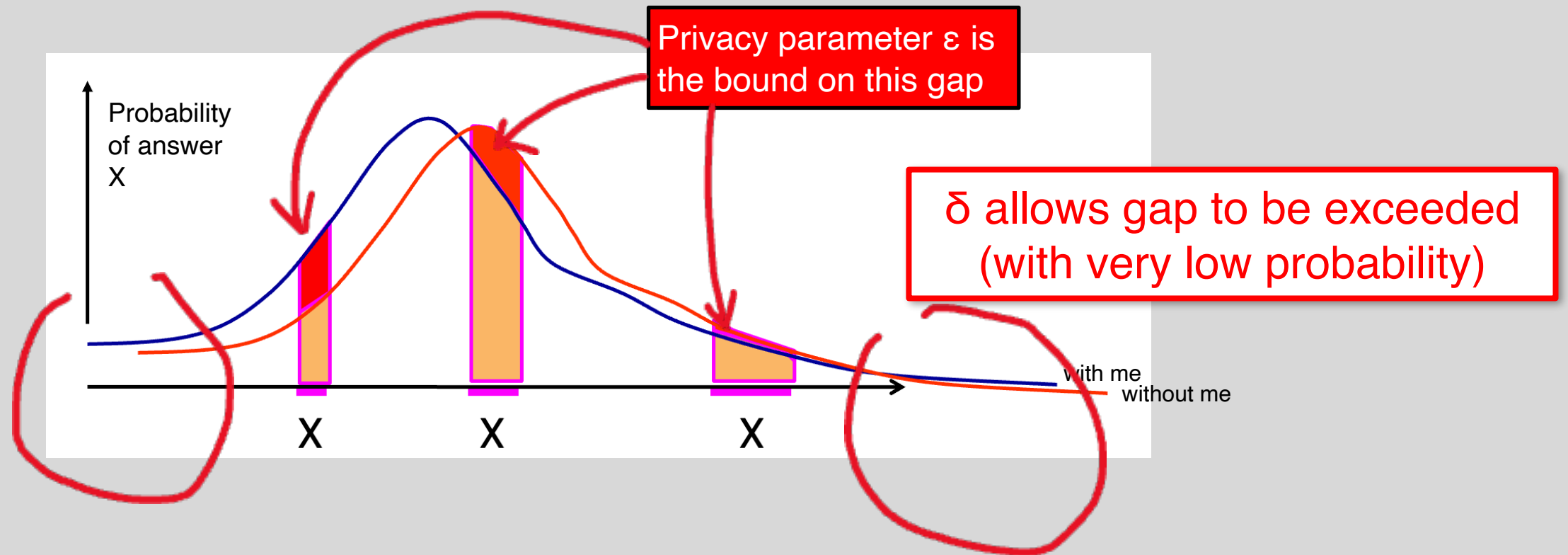
$$\Pr[A(x) \in S] \leq e^{\epsilon} \Pr[A(x') \in S] + \delta$$

Probability of
failure (ie, privacy
breach)

Want δ much smaller than
 $1/\text{poly}(N)$ -- why?

Intuition

Anything adversary can learn about me from the DB, it could learn without my data in the DB



DP-SGD ALGORITHM

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \dots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N} \sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t , noise scale σ , group size L , gradient norm bound C .

Initialize θ_0 randomly

for $t \in [T]$ **do**

 Take a random sample L_t with sampling probability L/N

Compute gradient

 For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C})$

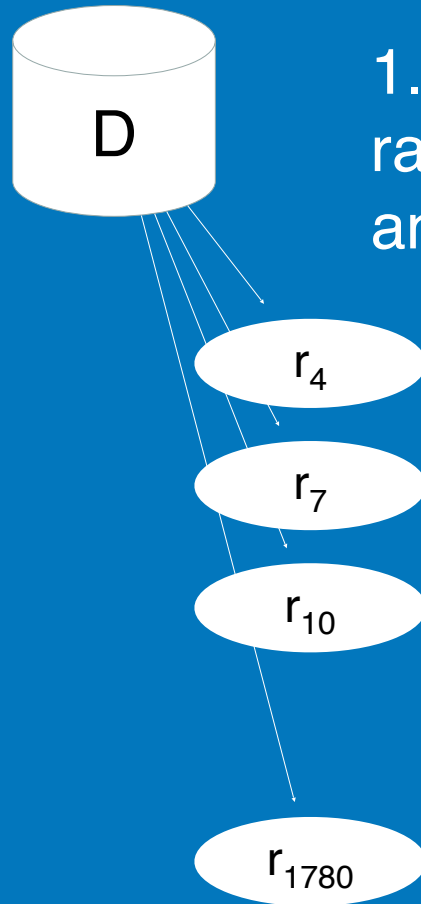
Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} (\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

Descent

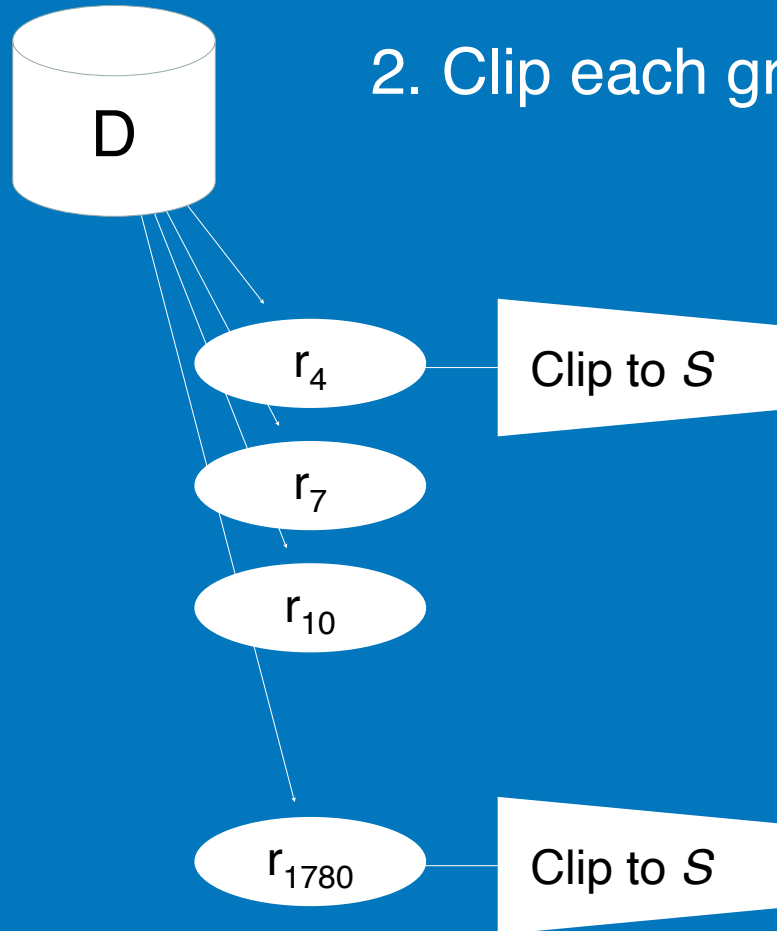
$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ϵ, δ) using a privacy accounting method.

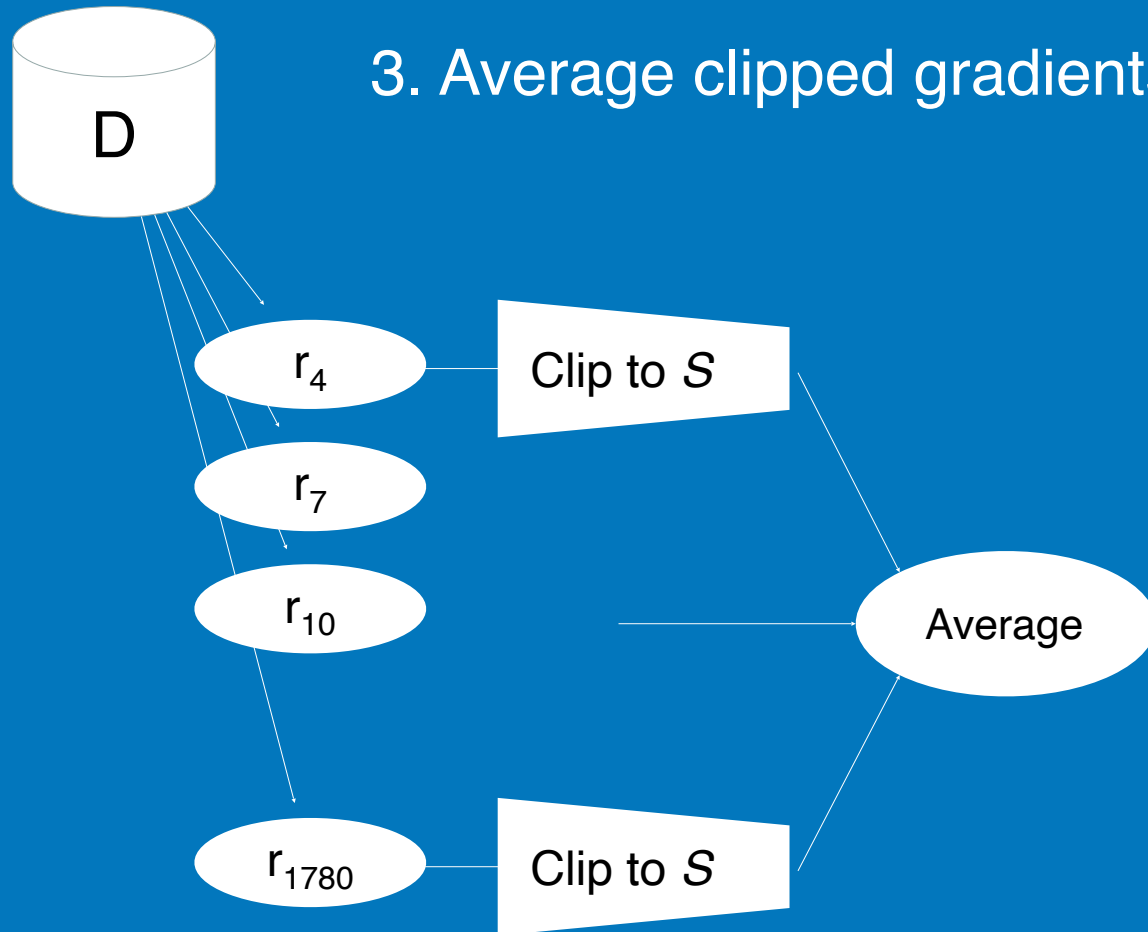


1. Sample a batch of examples uniformly at random,
and compute gradients for the current model

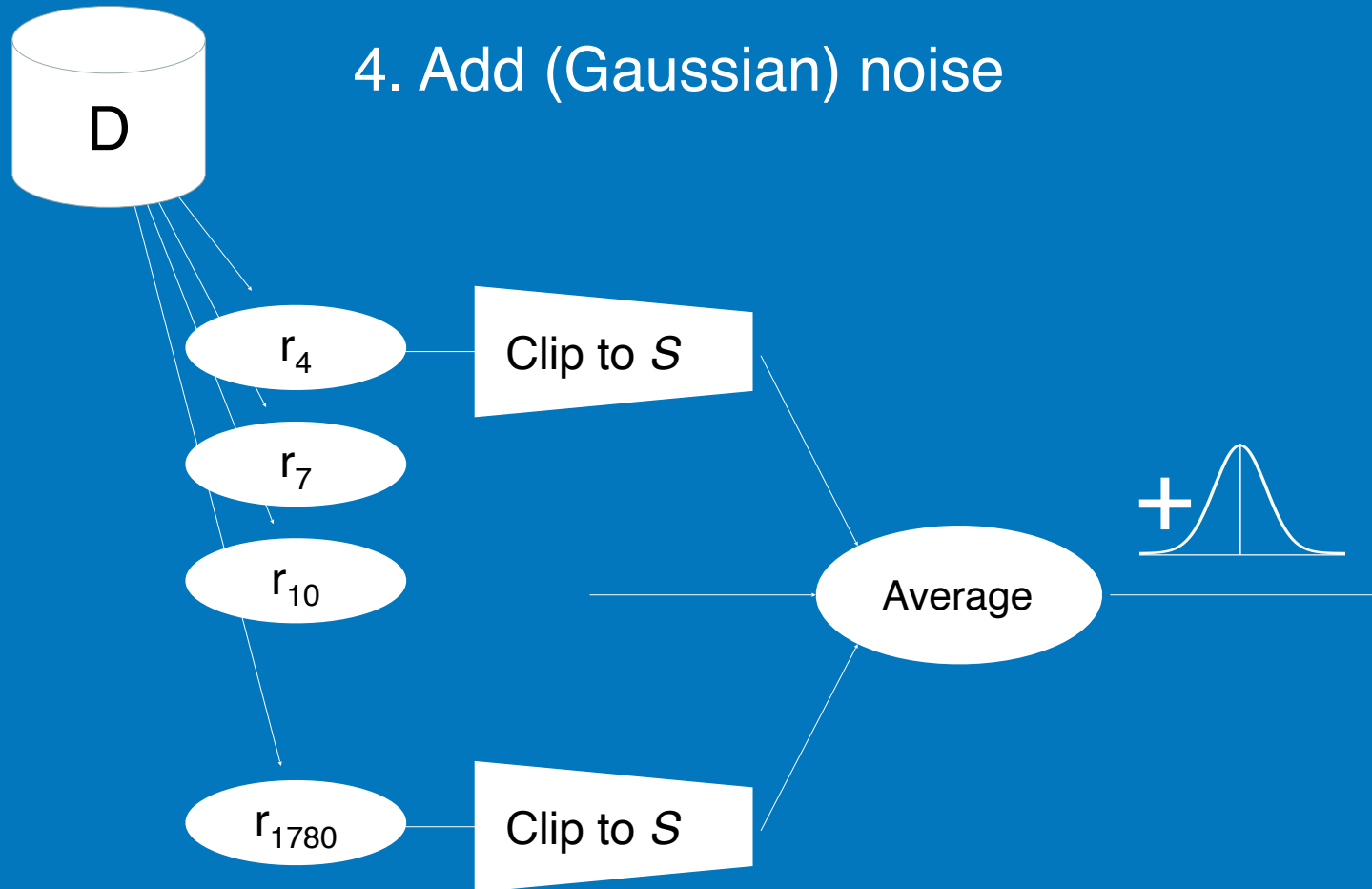
2. Clip each gradient to maximum L_2 norm S



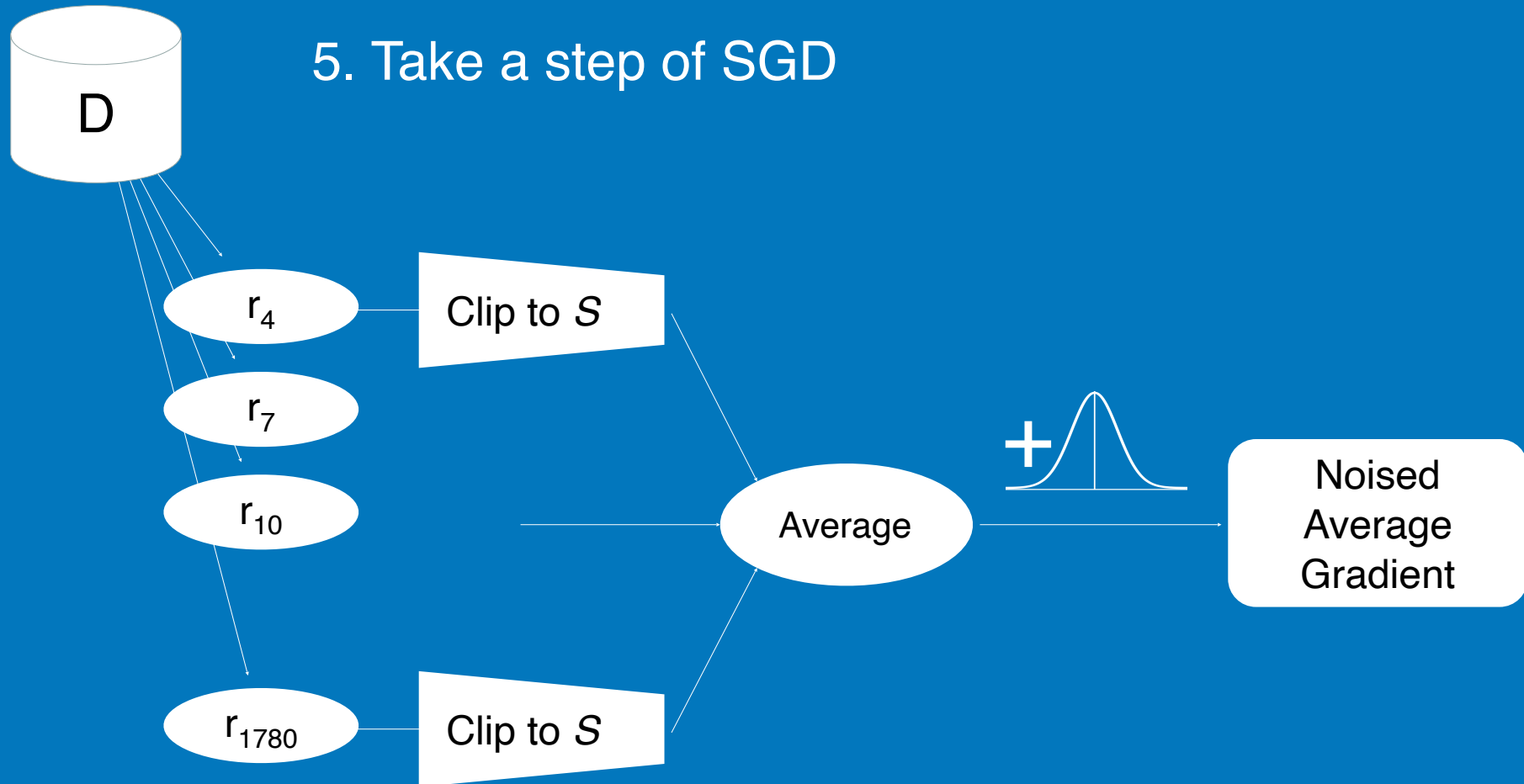
3. Average clipped gradients



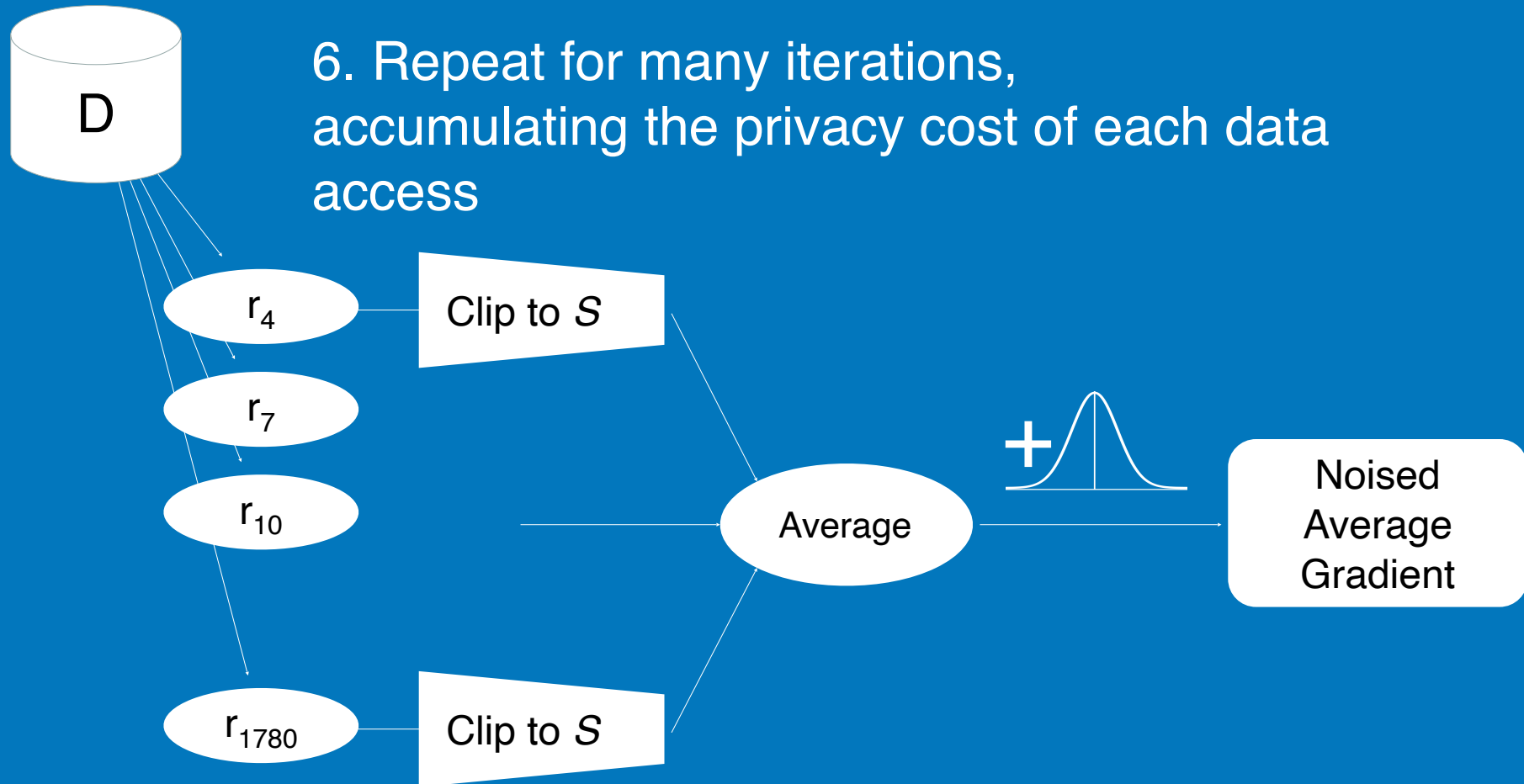
4. Add (Gaussian) noise



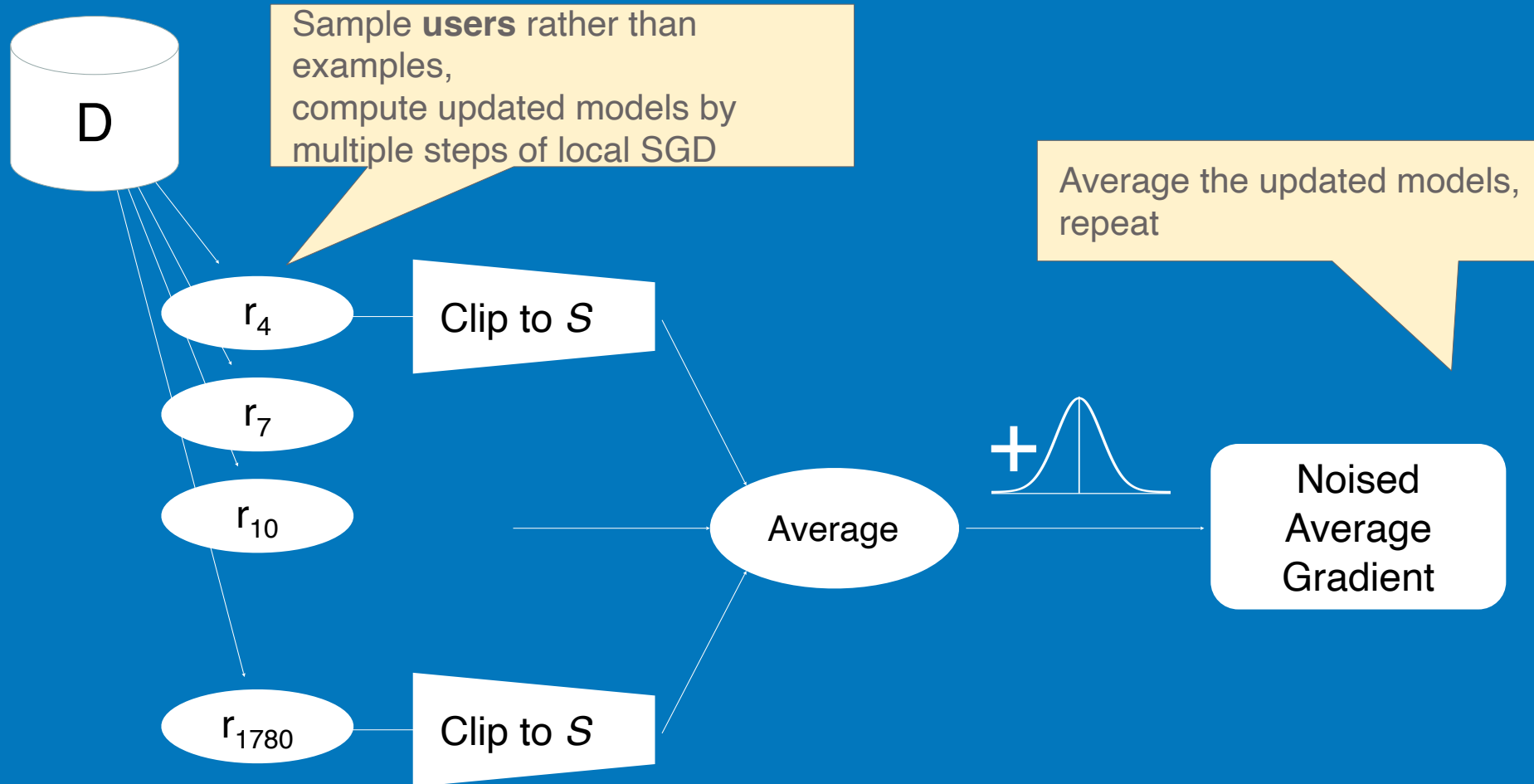
5. Take a step of SGD



6. Repeat for many iterations,
accumulating the privacy cost of each data
access



slide credit: Peter
Kairouz



Extension to **user-level** privacy

Clipping Gradients

If the gradient at step t by an input sample x_i is $\mathbf{g}_t(x_i) = \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$, the clipped gradient $\bar{\mathbf{g}}_t(x_i)$ is:

$$\bar{\mathbf{g}}_t(x_i) = \begin{cases} \mathbf{g}_t(x_i) & \text{if } \|\mathbf{g}_t(x_i)\|_2 \leq C \\ \frac{\mathbf{g}_t(x_i)}{\|\mathbf{g}_t(x_i)\|_2 / C} & \text{if } \|\mathbf{g}_t(x_i)\|_2 > C \end{cases}$$

- Limit influence of individual training inputs
- Bound sensitivity (important for calculating how much noise to add)

Adding Noise to Gradients

$$\tilde{\mathbf{g}}_t = \frac{1}{L} \left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right)$$

Average of clipped
gradients in a batch

Gaussian noise

Covariance depends on
clipping threshold C (why?)

Cumulative Privacy Loss

Each step has an (ϵ, δ) loss

A minibatch is sampled with probability $q = L/N$, thus $(q\epsilon, q\delta)$ privacy loss

T steps of gradient descent, thus **$(qT\epsilon, qT\delta)$** overall privacy loss



Scales linearly with T

Moments Accountant

Privacy loss of output o of computation $M^c(o; \mathcal{M}, \text{aux}, d, d') \triangleq \log \frac{\Pr[\mathcal{M}(\text{aux}, d) = o]}{\Pr[\mathcal{M}(\text{aux}, d') = o]}$

When composing multiple DP computations, cumulative privacy loss $<$ tail bound on the sum of the privacy loss variables at each step

It is sufficient to bound all **moments** of the computation at each step

n^{th} moment of a random variable X is $E(X^n)$

For Gaussian noise with random sampling, can derive a relatively tight moments bound

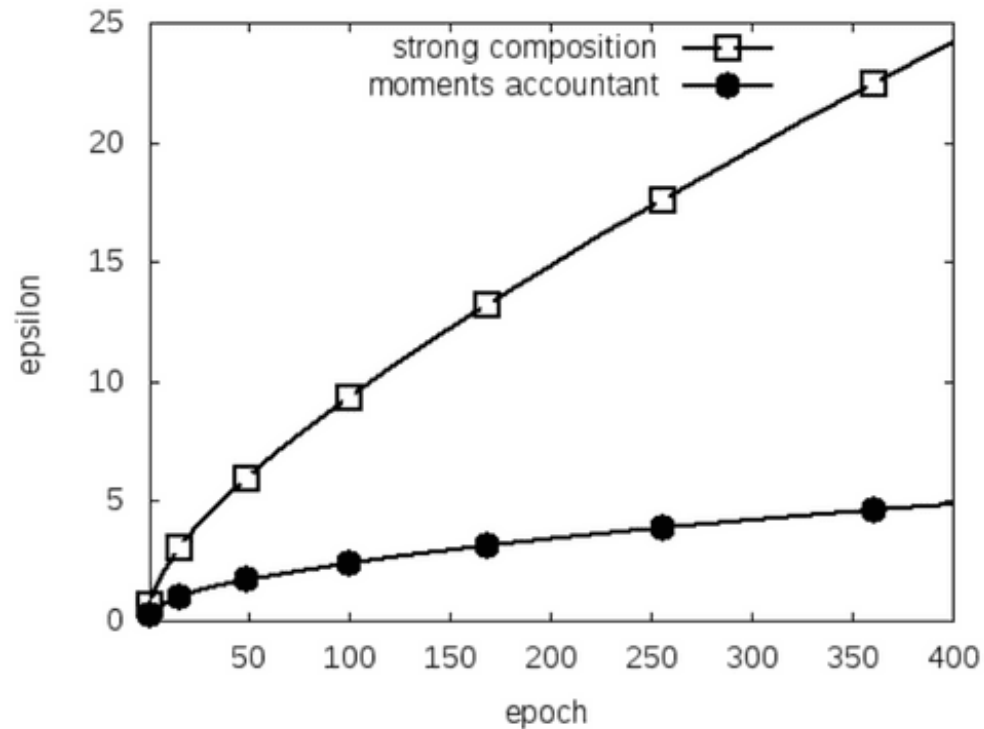


Figure 2: The ϵ value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.

Using the moments accountant, the overall training process consisting of T steps is

$O(q\epsilon\sqrt{T})$, δ -differentially private

Privacy loss is proportional to \sqrt{T} , not T

δ (probability of failure) is independent of T

pytorch/**opacus**

Training PyTorch models with differential privacy



Implements (ϵ) -Renyi differential
privacy

Renyi divergence:

$$D_{\alpha}(P\|Q) \triangleq \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left(\frac{P(x)}{Q(x)} \right)^{\alpha}$$

$$D_1(P\|Q) = \mathbb{E}_{x \sim P} \log \frac{P(x)}{Q(x)} \quad \leftarrow \text{KL divergence, aka relative entropy}$$

$$D_{\infty}(P\|Q) = \sup_{x \in \text{supp } Q} \log \frac{P(x)}{Q(x)}$$

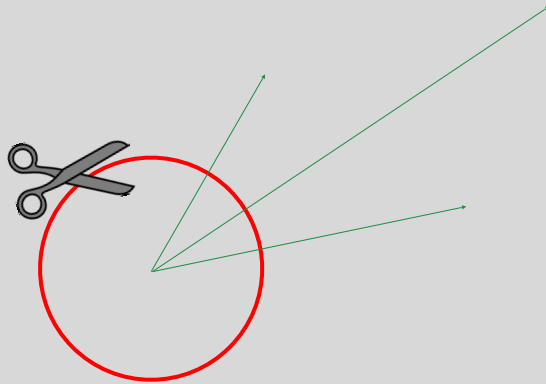
$$D_{\infty}(f(D)\|f(D')) \leq \epsilon$$

Renyi differential privacy

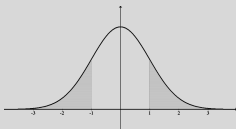
How to Clip in Practice?

Too small:

update magnitude lost,
poor learning

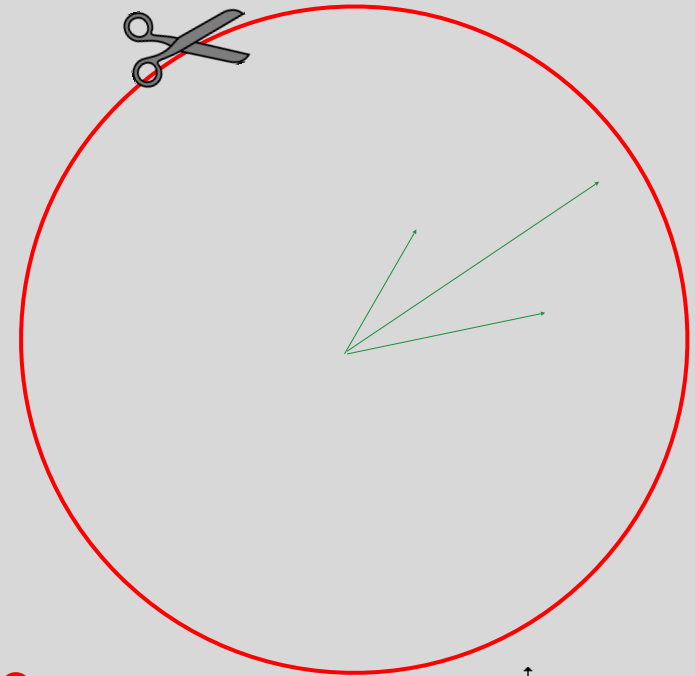


+

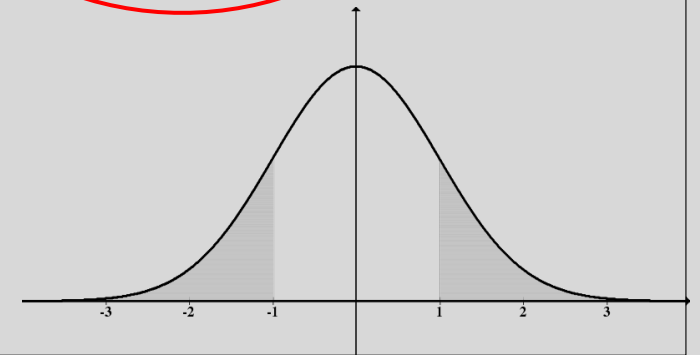


Too big:

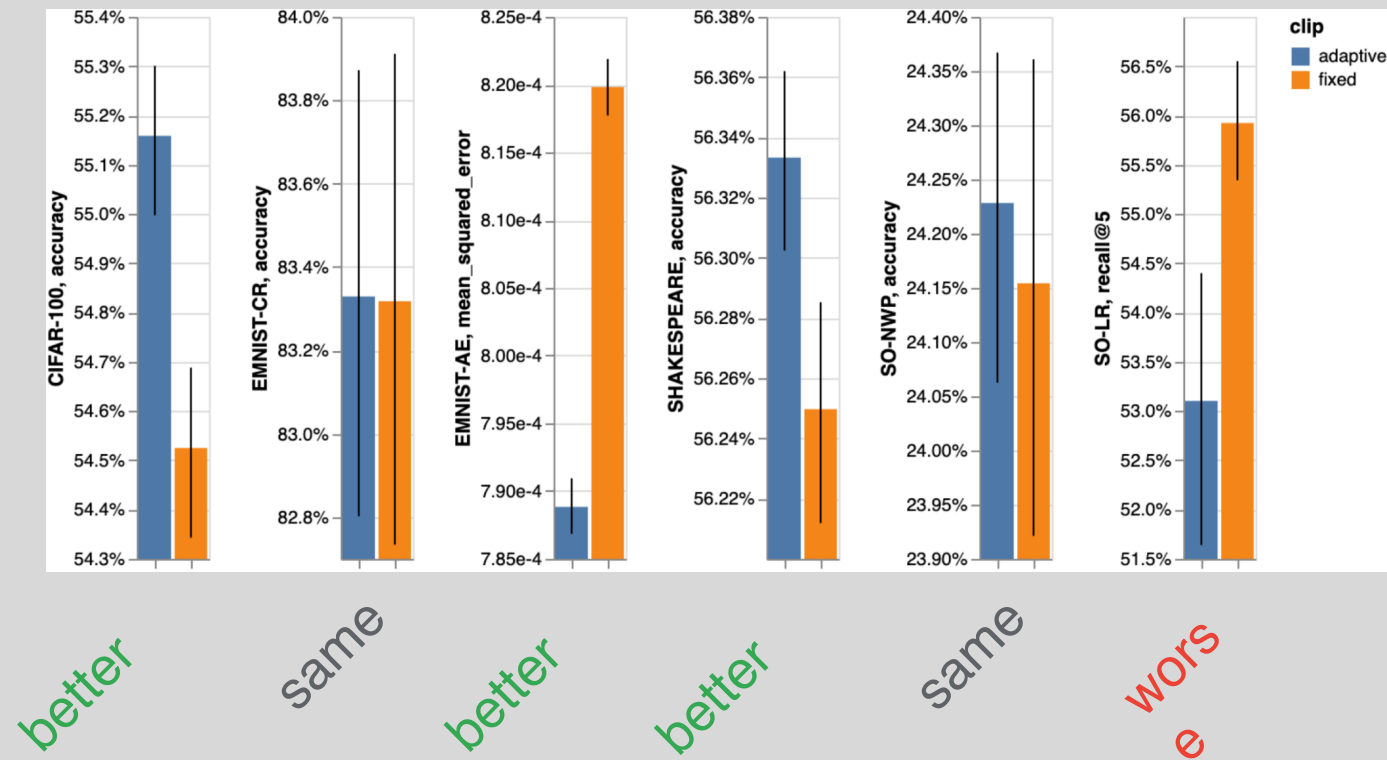
too much DP
noise needs to be
added



+



Adaptive Clipping to Median



Adaptive clipping to median with no hyperparameter tuning usually performs at least as well as best fixed clip **chosen in hindsight**

[Home](#) > [Blog](#) >

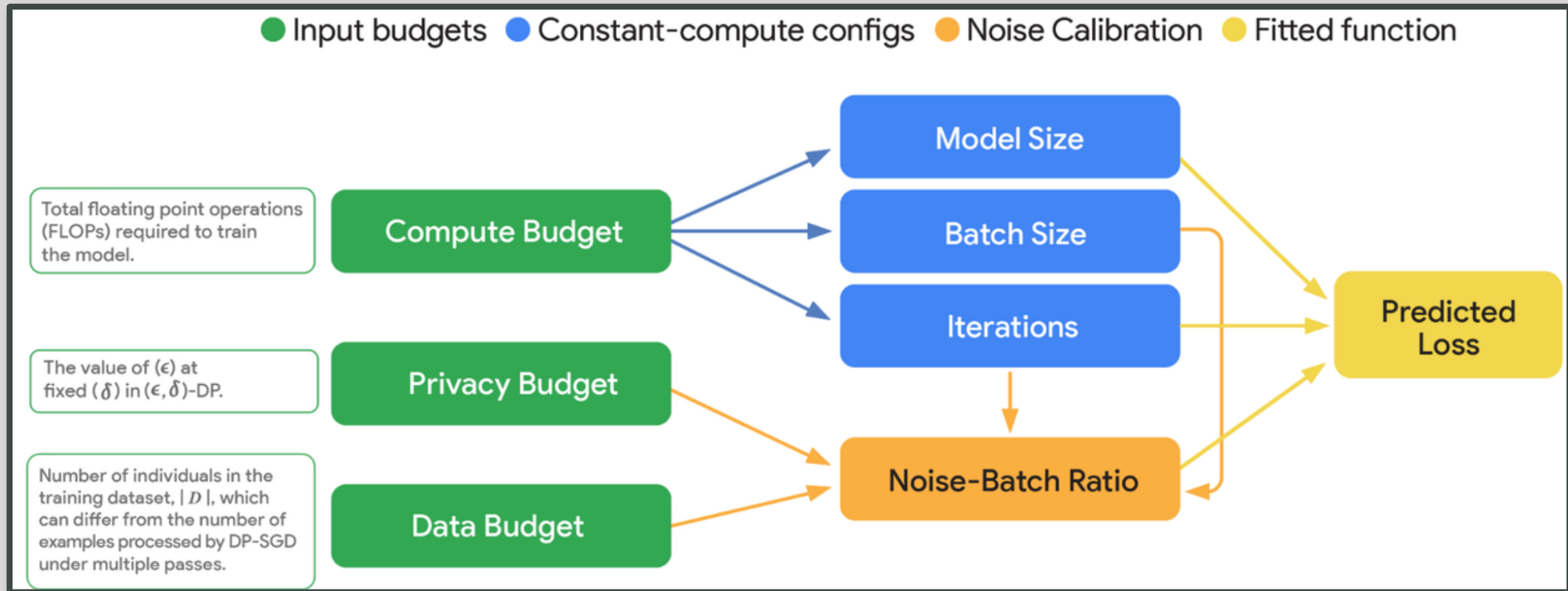
VaultGemma: The world's most capable differentially private LLM



September 12, 2025 ·

Amer Sinha, Software Engineer, and Ryan McKenna, Research Scientist, Google Research

DP Scaling Laws



<https://research.google/blog/vaultgemma-the-worlds-most-capable-differentially-private-llm/>

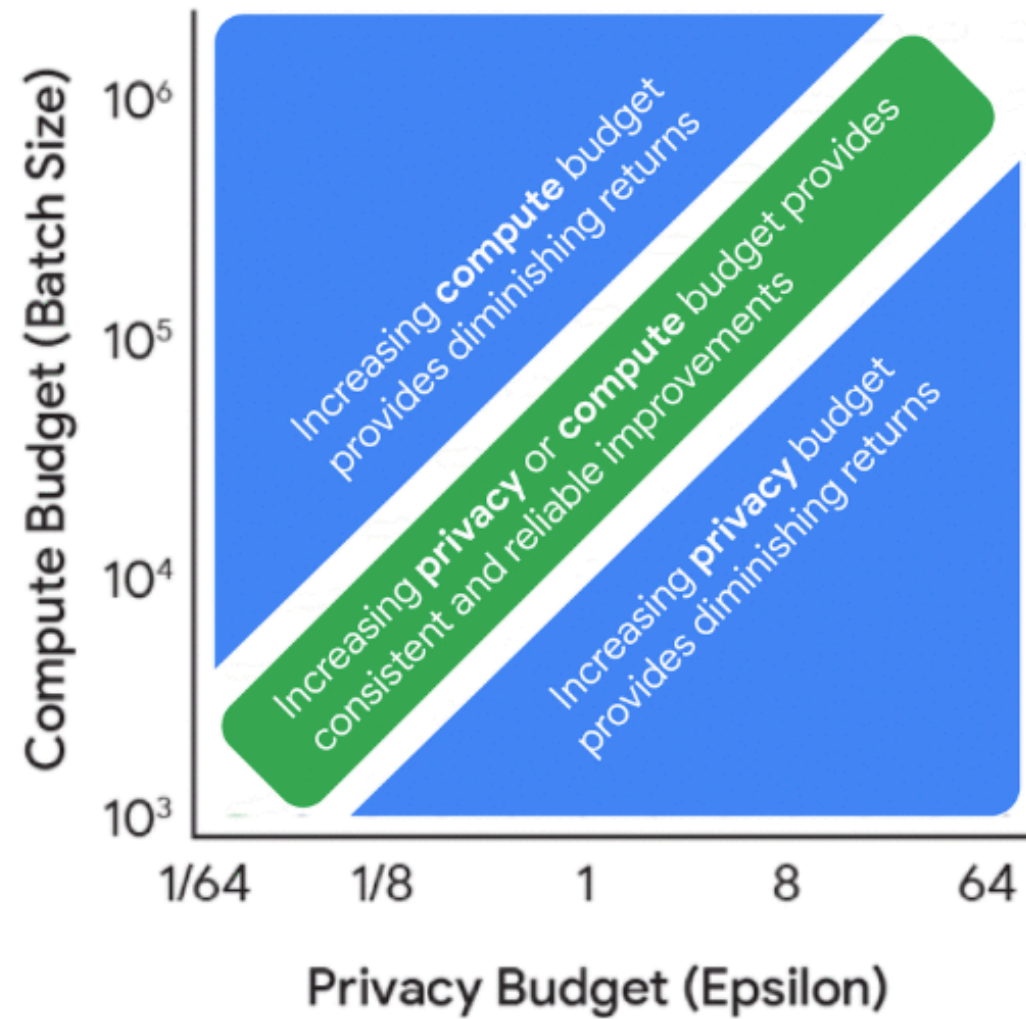
Key Technical Innovation: Better Batching!

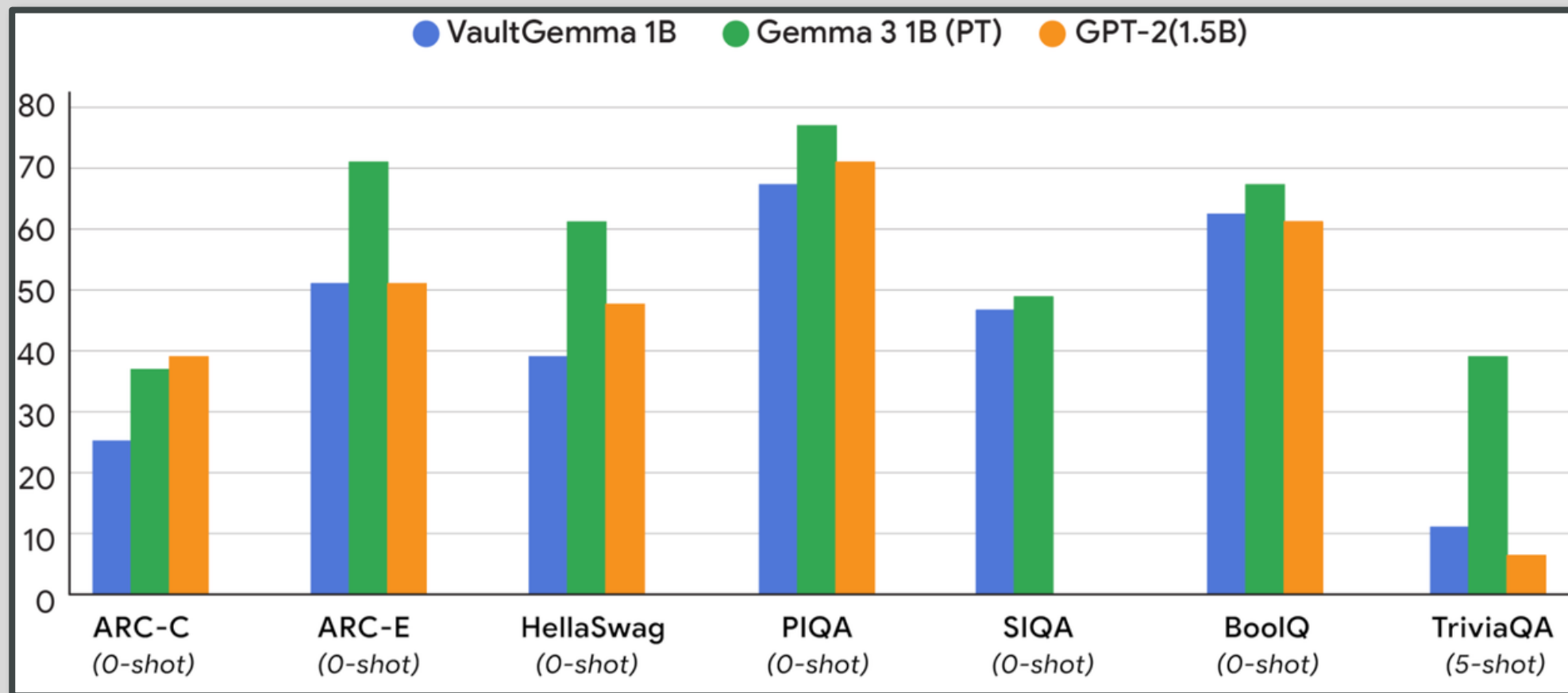
Poisson sampling: instead of sampling fixed-size batches, sample each training input with a fixed probability

- Better privacy analysis but produces variable-size batches

Truncated Poisson subsampling to bound batch size

Massively parallel computation (MapReduce, etc.) for scalable sampling





SOTA DP LLM of 2025 is comparable to a non-private LLM from 2019

<https://research.google/blog/vaultgemma-the-worlds-most-capable-differentially-private-llm/>

Unanswered Questions

What is the right “unit” of privacy

- A sequence of tokens? How long? All data from a given individual? ... “about” an individual? What’s an “individual”? Duplicated or shared data?

What is the right value of epsilon?

How to translate this epsilon into understandable privacy risks?

What is the right privacy-utility tradeoff for a given use?