# CSCI 699: Privacy Preserving Machine Learning - Week 5

## Gaussian DP and Privacy Auditing

Sai Praneeth Karimireddy, Sep 27 2024

# Recap

- Composition: simple - $k\varepsilon$-DP

**Theorem. Advanced Composition**

A combination of $A_1 \circ A_2 \circ A_k$, each of which is $(\varepsilon, \delta)$-DP is $(\tilde{\varepsilon}, \tilde{\delta})$-DP where

$$\tilde{\varepsilon} = \varepsilon\sqrt{2k\ln(1/\delta')} + k\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} \quad \text{and} \quad \tilde{\delta} = k\delta + \delta'$$

For any choice of $\delta'$.

# Recap

- Subsampling amplification

**Theorem. Subsampling Amplification**

Composing an $(\varepsilon, \delta)$-DP A with a sampling rate of $q$ results in an $(\tilde{\varepsilon}, \tilde{\delta})$-DP algorithm where

$$\tilde{\varepsilon} = \log(1 - q + qe^{\varepsilon}) = O(q\varepsilon) \quad \text{and} \quad \tilde{\delta} = q\delta$$

# Recap

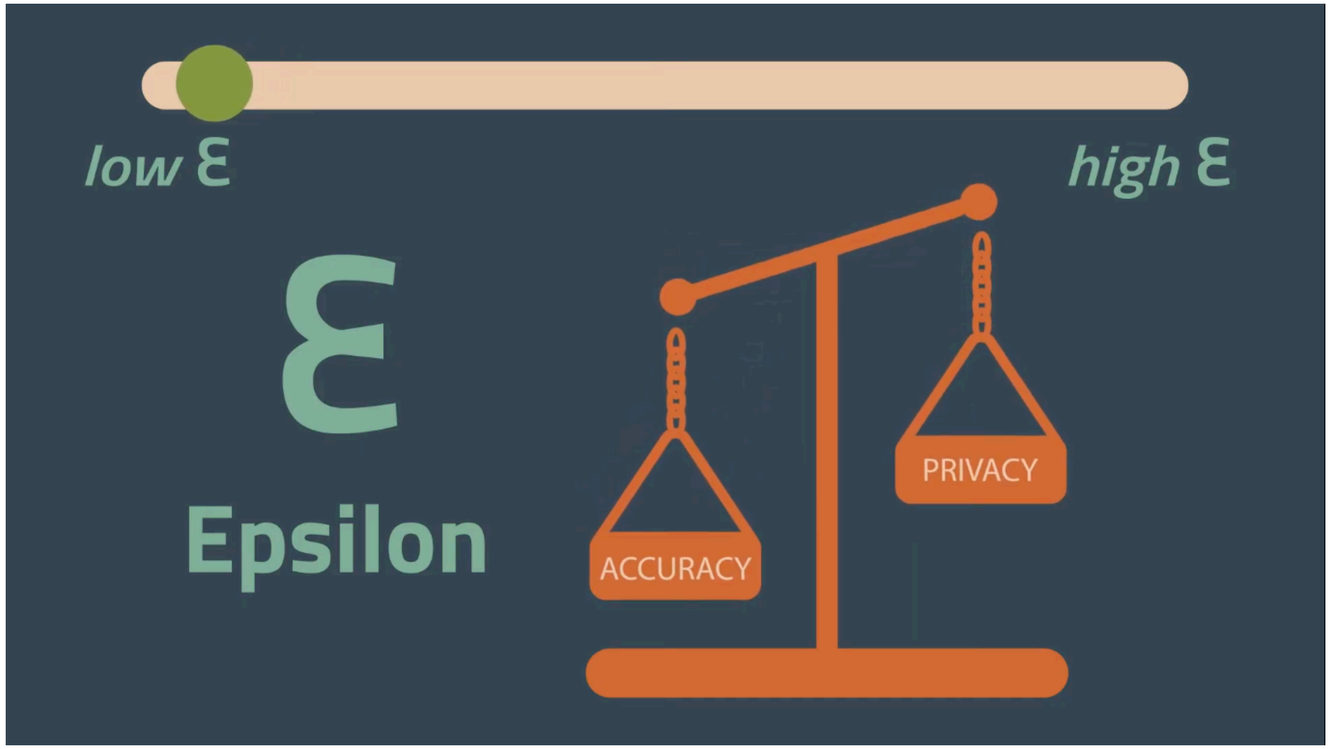
- Private SGD with clipping L1 norm:
  - $\theta_t = \theta_{t-1} - \gamma \text{Clip}_\tau \left( \nabla_\theta \ell(f(x_t; \theta), y_t) \right) + Lap(2\tau/\varepsilon)$
- With $q = 1/n$, k rounds satisfies $(O(\varepsilon/n\sqrt{k \ln(1/\delta)}), \delta)$-DP for any $\delta > 0$.
- Can also clip L2 norm and use Gaussian mechanism.
- Q: what did you observe empirically L1 vs. L2?

# Agenda for today

## Analyzing privacy of ML training

- Gaussian DP
- Privacy Auditing
- Presentations + discussions
- Auditing Practical - next week (needs HW2 soln)

# Gaussian Differential Privacy



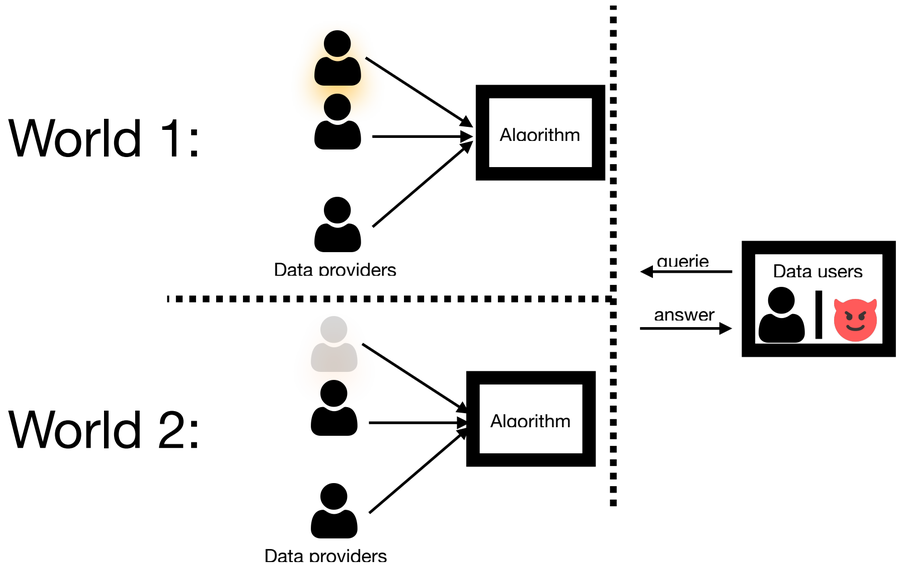
# Drawbacks of Approximate DP

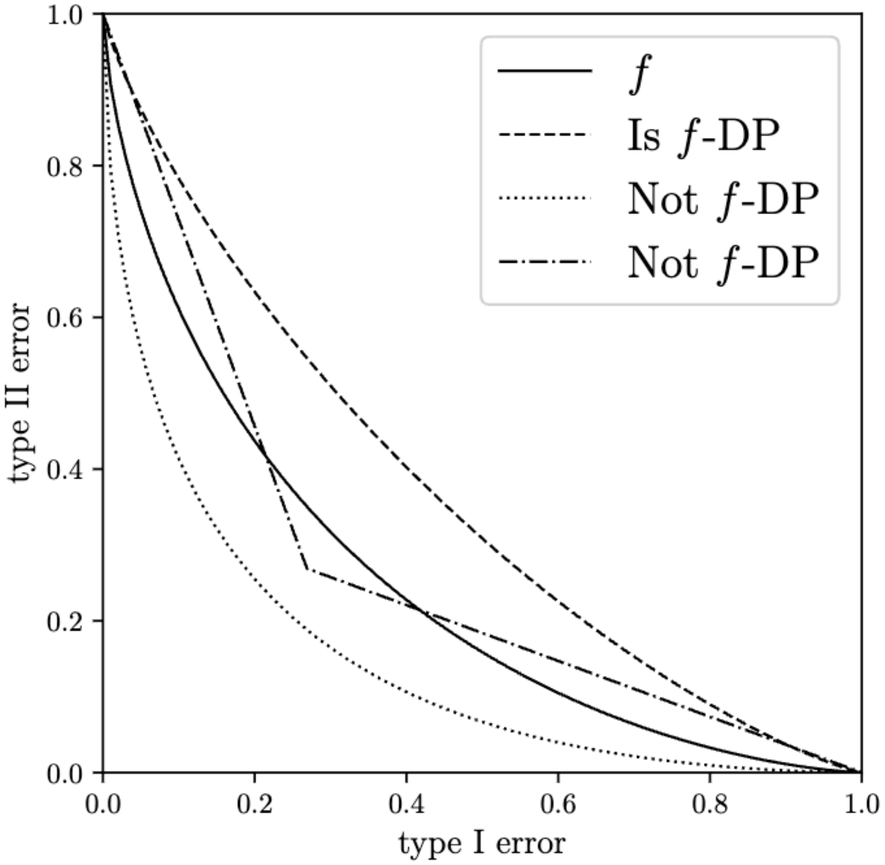
- After k steps of Lap-SGD, we were able to show $(\varepsilon\sqrt{2k\ln(1/\delta)}, \delta)$-DP
- But advanced composition is too lose.

# f-DP

## Most general privacy definition

- **Definition.** Given a function $f$, we say an algorithm is $f$-DP if the tradeoff curve of an optimal distinguisher is strictly above f.



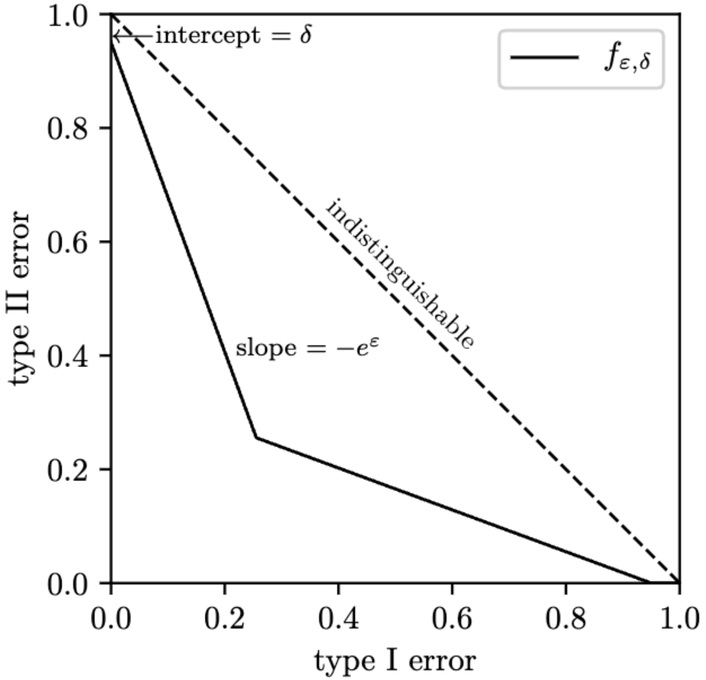



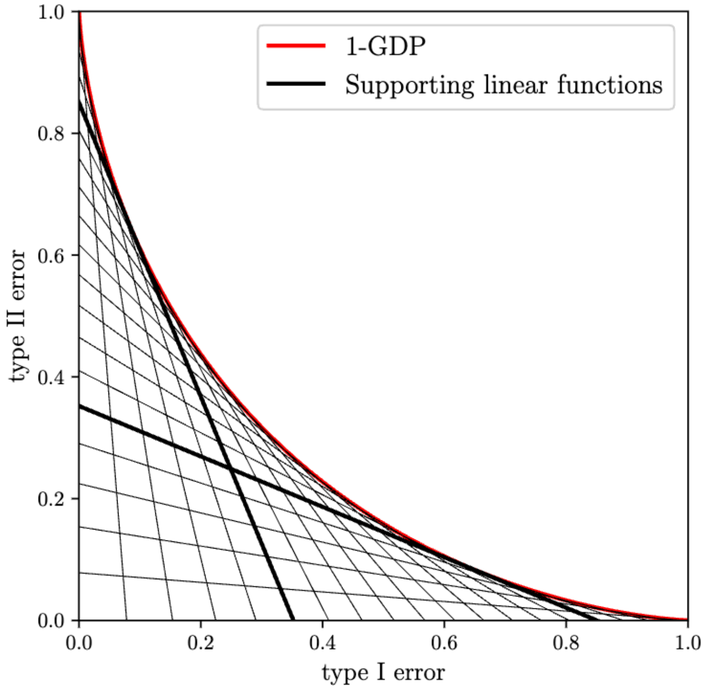
# f-DP

## Generalization $(\varepsilon, \delta)$-DP

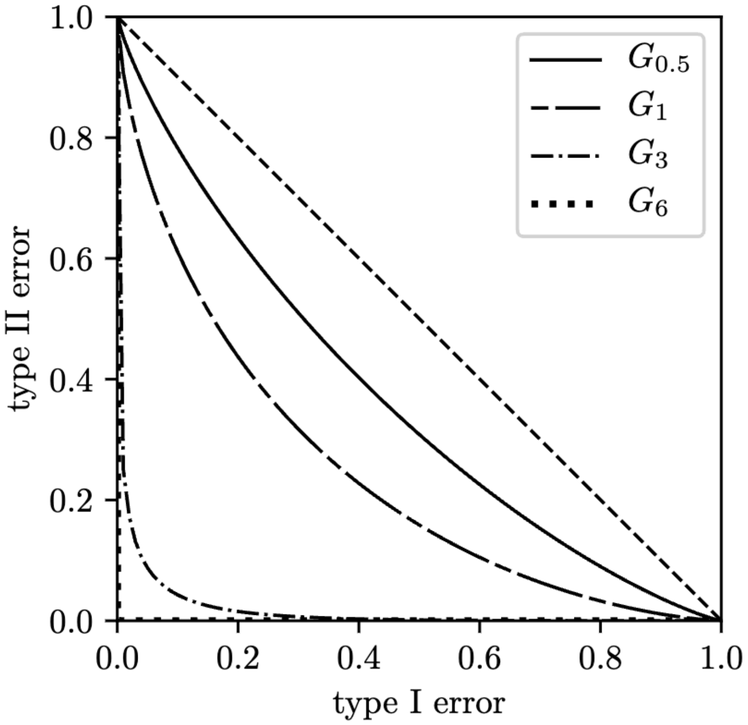
- **Prop 2.5** [WZ10]**.** A is $(\varepsilon, \delta)$-DP iff it satisfies $f_{\varepsilon,\delta}$-DP for
  $f_{\varepsilon,\delta} = \max(1 - \delta - e^{\varepsilon}x \ , \ (1 - \delta - x)/e^{\varepsilon})$

- **Prop 2.12** [DRS19] A is $f$-DP iff it satisfies $(\varepsilon, \delta_f(\varepsilon))$-DP for $\forall \varepsilon \geq 0$ and
  $\delta_f(\varepsilon) = 1 + f^*(-e^{\varepsilon})$.

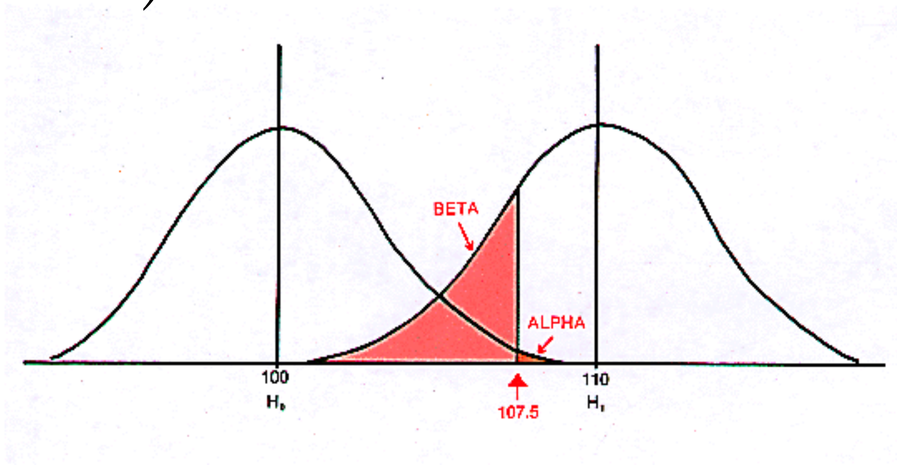


# Gaussian-DP

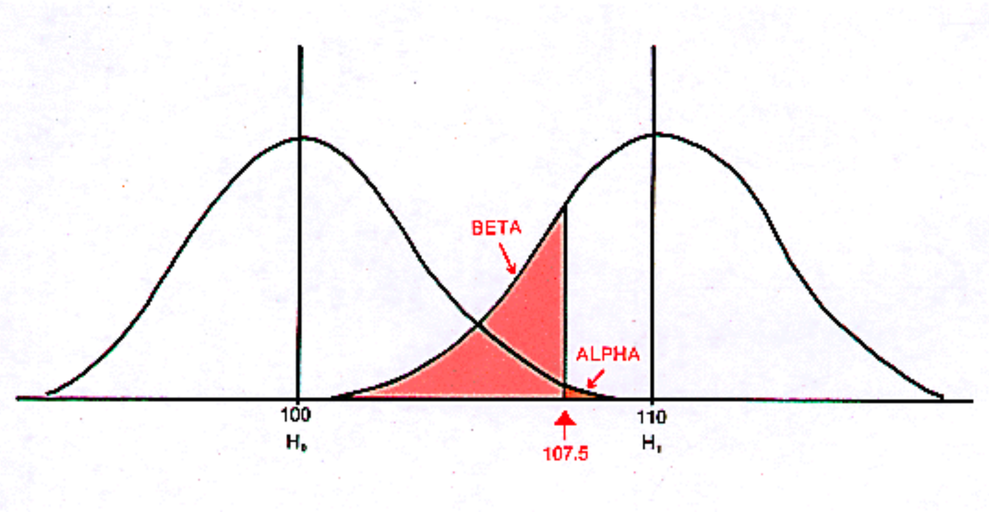



- **Definition.** A is $\mu$-GDP if it satisfies $f_\mu$-DP for $f_\mu = T\left(\mathcal{N}(0,1)\ ,\ \mathcal{N}(\mu,1)\right)$
- $$\frac{Pr[A(D)=t]}{Pr[A(D')=t]} \leq \frac{Pr[\mathcal{N}(0,1)=t]}{Pr[\mathcal{N}(\mu,1)=t]} = \exp\left(\frac{1}{2}(\mu^2 - 2\mu t)\right)$$
- $\alpha(\tau) = 1 - \Phi(\tau)$ and $\beta(\tau) = \Phi(\tau - \mu)$



# Gaussian-DP

## Gaussian mechanism




- **Definition.** A is $\mu$-GDP if it satisfies $f_\mu$ -DP for $f_\mu = T\left(\mathcal{N}(0,1)\ ,\ \mathcal{N}(\mu,1)\right)$

**Theorem. Gaussian mechanism**

Given $f : \mathcal{X}^n \to \mathbb{R}^d$ with $\Delta$ bounded $\ell_2$-sensitivity, $f(D) + \mathcal{N}\left(0\ ,\ \frac{\Delta^2}{\mu^2} I_d\right)$ is $\mu$-GDP.

# Gaussian Differential Privacy

## Tight composition

**Theorem. GDP Composition**

Composition of $A_1 \circ A_2 \ldots \circ A_k$, each of which is $\mu_i$ -GDP is $\sqrt{\sum_{i=1}^{k} \mu_i^2}$-GDP.

# Gaussian Differential Privacy

## Canonical f

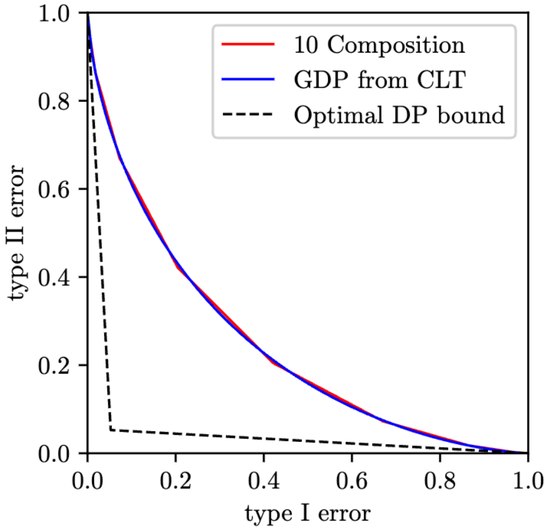
**Theorem 3.4 [DRS19] Central limit theorem of composition**

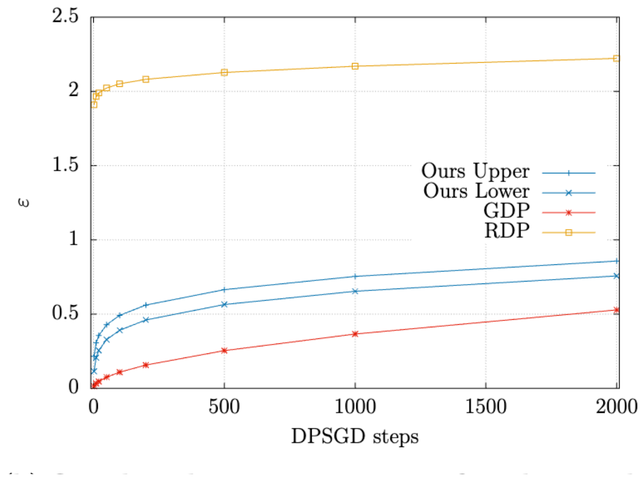
Given some regularity assumptions, composition of $A_1 \circ A_2 \ldots \circ A_k$, each of which is $f_i$-DP is approximately $\mu$-GDP for

$$\mu = \frac{2\sqrt{k}\kappa_1}{\kappa_1 - \kappa_2} \text{ for } \kappa_1 = -\int_0^1 \log|f'(x)|\,dx \text{ and } \kappa_2 = -\int_0^1 \log^2|f'(x)|\,dx.$$

# Gaussian Differential Privacy

## Canonical f



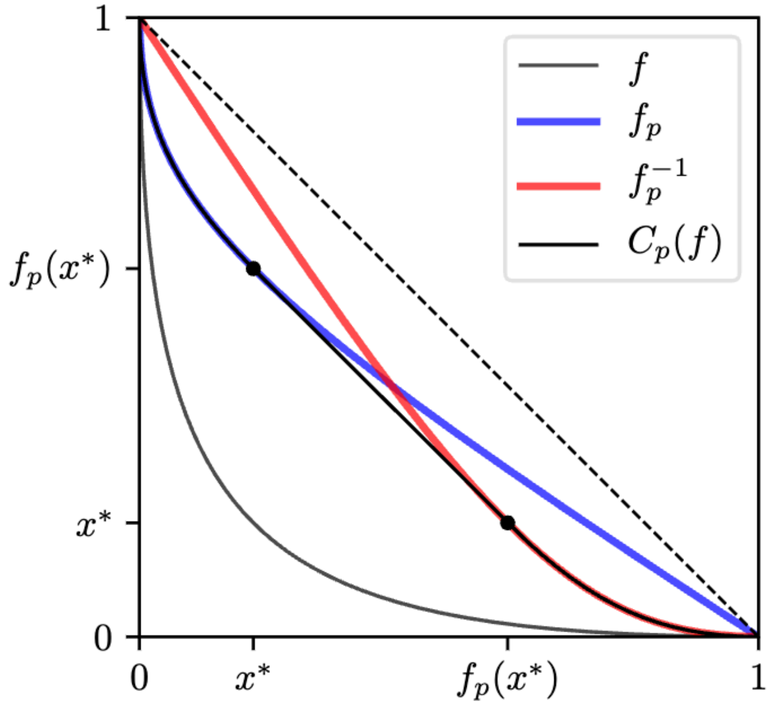




- In stats, combining may random variables ≈ Gaussian by CLT. In DP, composing many DP steps ≈gDP.

- Caution: just like CLT sometimes fails, Thm 3.4 is sometimes fails and underestimates privacy [GLW21].

# Gaussian Differential Privacy

## Amplification by subsampling




- Define $f_q(x) = qf(x) + (1-q)(1-x)$ and $f_q^{-1}$
- **Theorem 4.2** [DRS19]
  Composing q-sampling with $f$-DP, is $\left(\min(f_p, f_p^{-1})\right)^{**}$-DP
- Unfortunately, no closed form for GDP, compute numerically.

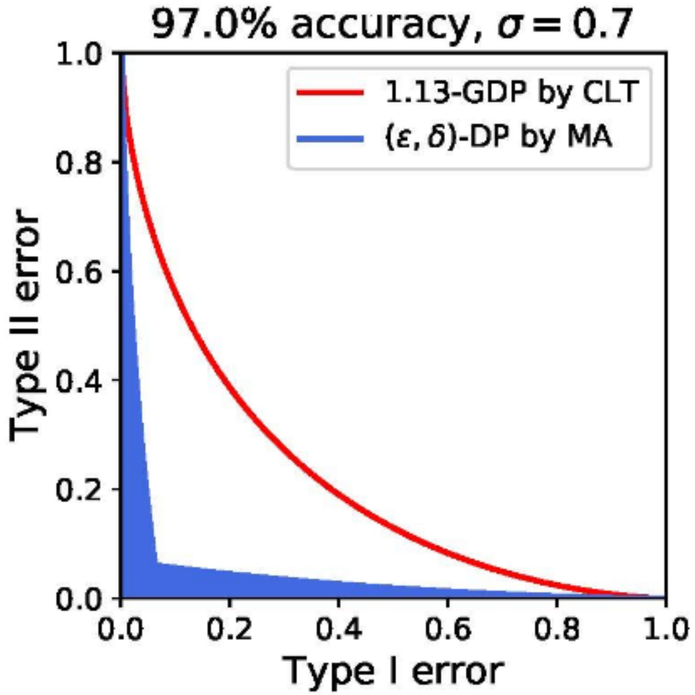
# Private SGD

## Using Gaussian-DP

**Corollary 5.4 [DRS19] Subsampled Composition**

Suppose each $A_i$ is $\mu$-GDP. Then, composing q-sampled $A_i$ is asymptotically

$$\left(q\sqrt{k}\sqrt{e^{\mu^2}\Phi(3\mu/2)+3\Phi(-\mu/2)-2}\right)\text{-GDP.}$$




Tightest privacy bound [B+'20]. But, only asymptotically valid.

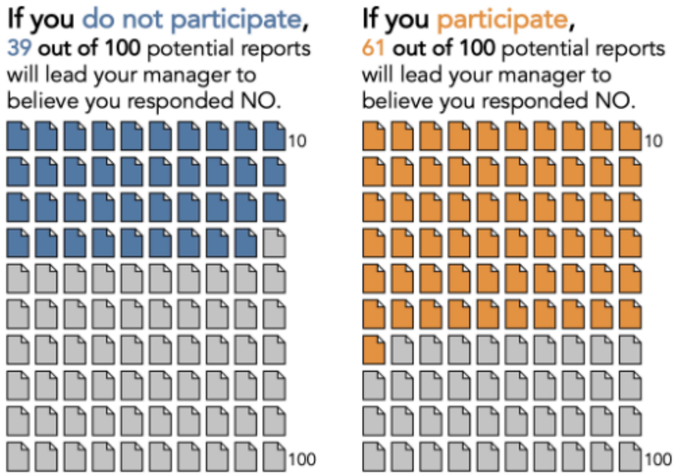
# Aside: Communicating Privacy

## Odds ratio

If you **do not participate**,
39 out of 100 potential reports will lead your manager to believe you responded NO.

If you **participate**,
61 out of 100 potential reports will lead your manager to believe you responded NO.

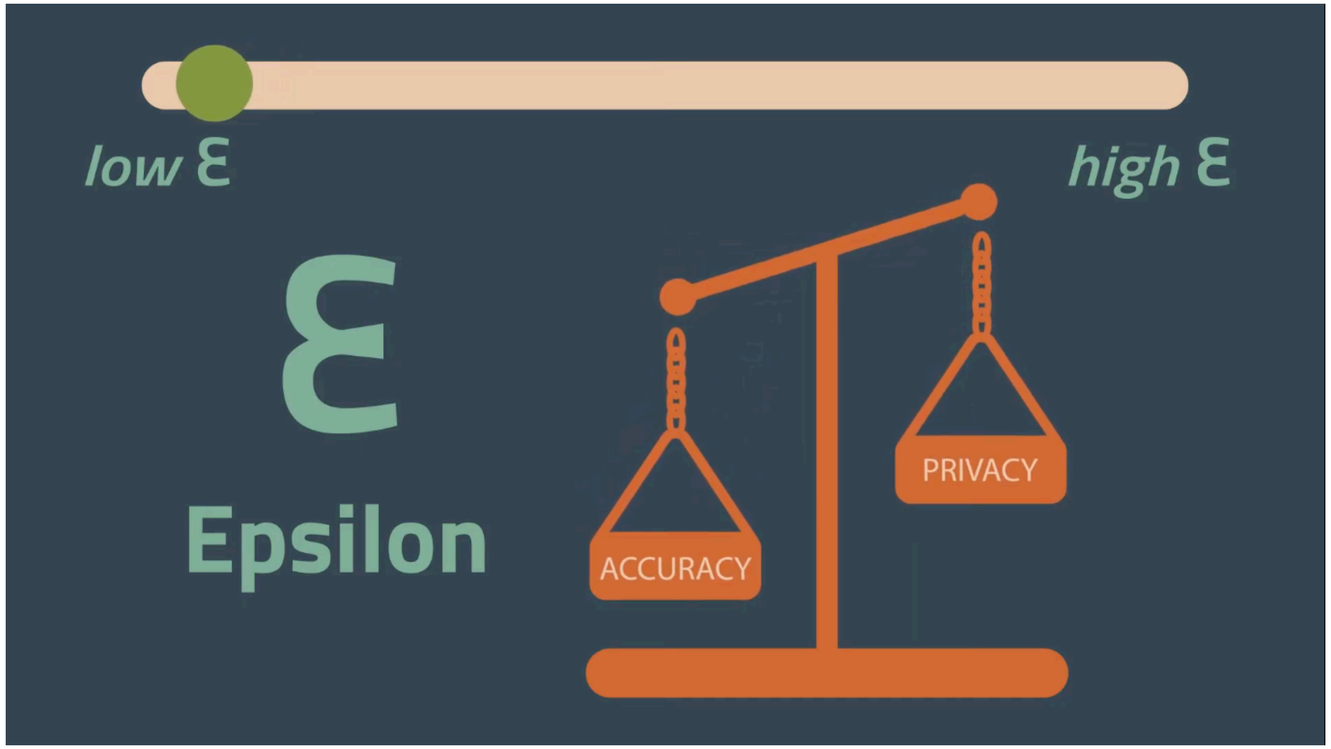
(a) ODDS-TEXT




(b) ODDS-VIS

- How do you communicate privacy risk to your friends?
- Excellent study: [N+UseNIX'23]
- Using odds ratio leads to increased understanding of risks and willingness to share data.
- How to explain $\varepsilon$-DP and $\mu$-GDP? Need to incorporate prior knowledge of attacker.

# Privacy Auditing



# Drawbacks of pure theory

- Bounds always loose
    - people assume this and train models with high theoretical $\varepsilon$
- Maybe my implementation is incorrect
- Why should I trust your claim?

**Backpropagation Clipping for Deep Learning with Differential Privacy**

Timothy Stevens*
*University of Vermont*

Ivoline C. Ngong*
*University of Vermont*

David Darais
*Galois, Inc.*

Calvin Hirsch
*Two Six Technologies*

David Slater
*Two Six Technologies*

Joseph P. Near
*University of Vermont*

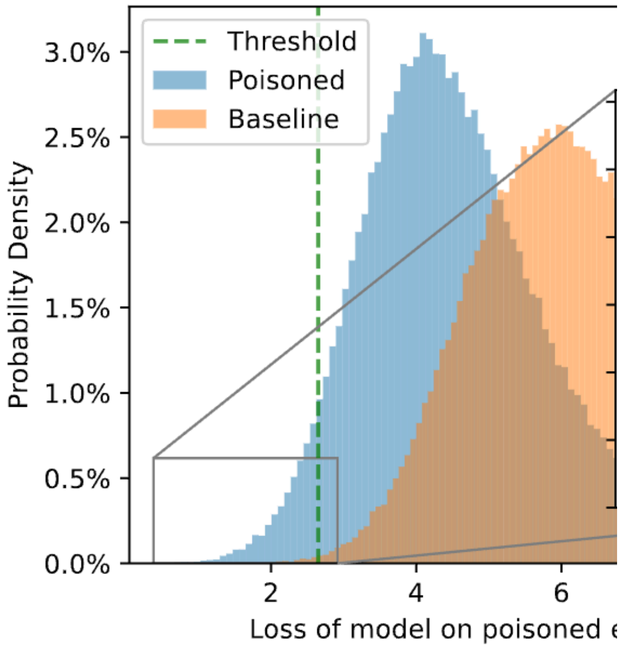
- In 2022, proposed to integrate clipping into forward/backward pass directly
- SOTA accuracy with 30x smaller $\varepsilon$

# Privacy Auditing

**Debugging Differential Privacy: A Case Study for Privacy Auditing**

*Florian Tramèr*, Andreas Terzis, Thomas Steinke, Shuang Song, Matthew Jagielski, Nicholas Carlini*
*Google Research*

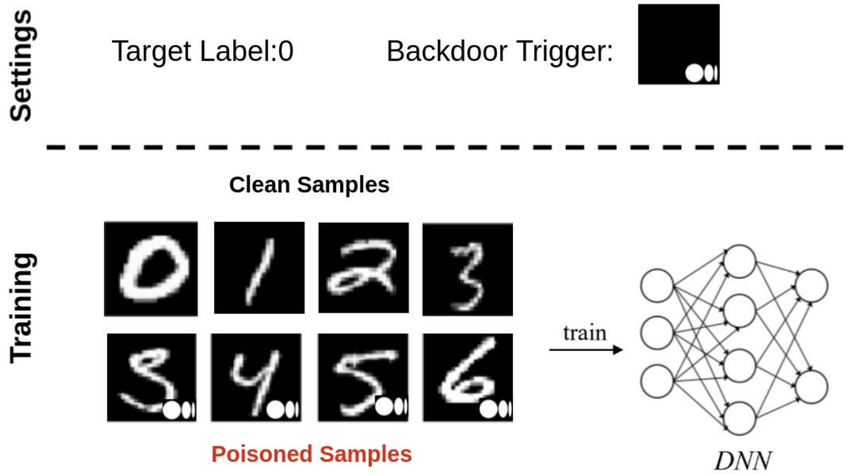
- Consider the following test:
  - D = MNIST dataset: 60k images
  - D' = Add $(x', y')$.
  - Train a CNN $\theta$ using [S+22] to get 0.98 acc and (0.21, 10–5)-DP.
  - Check $\ell_\theta(x', y') \leq \tau$. If D' will be smaller.
  - Repeat 100k on D and 100k on D'.

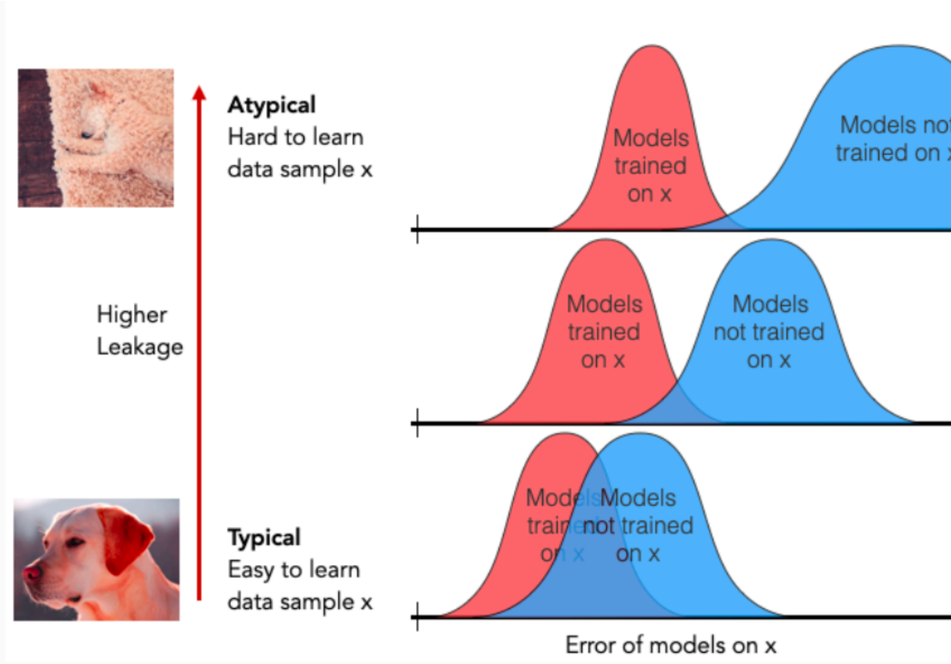


# Privacy Auditing

- Some decisions to make

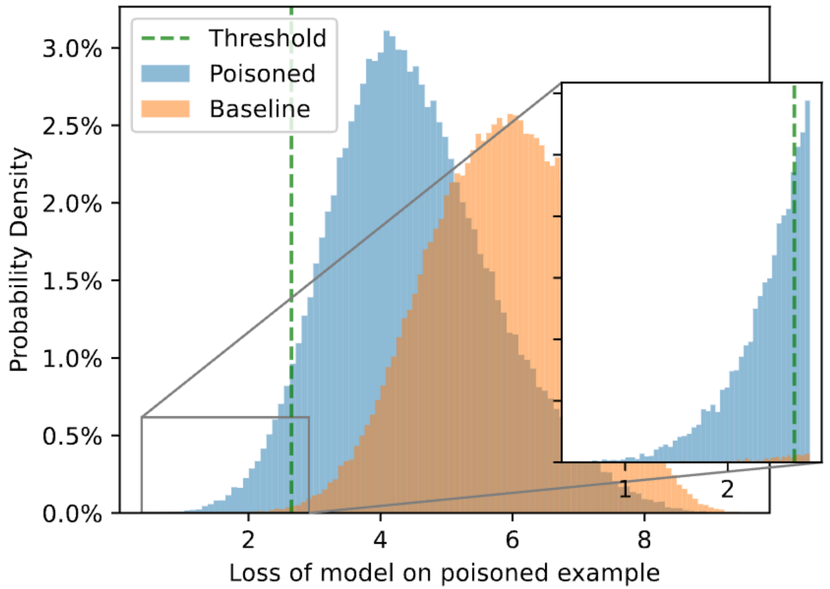
  - Which $x'$? Called canary
  - insert an *unique* image which model is likely to memorize. i.e. insert a *backdoor attack*
  - What $y'$? Any incorrect label
  - Try a few images (~25) on an initial 2k training runs.



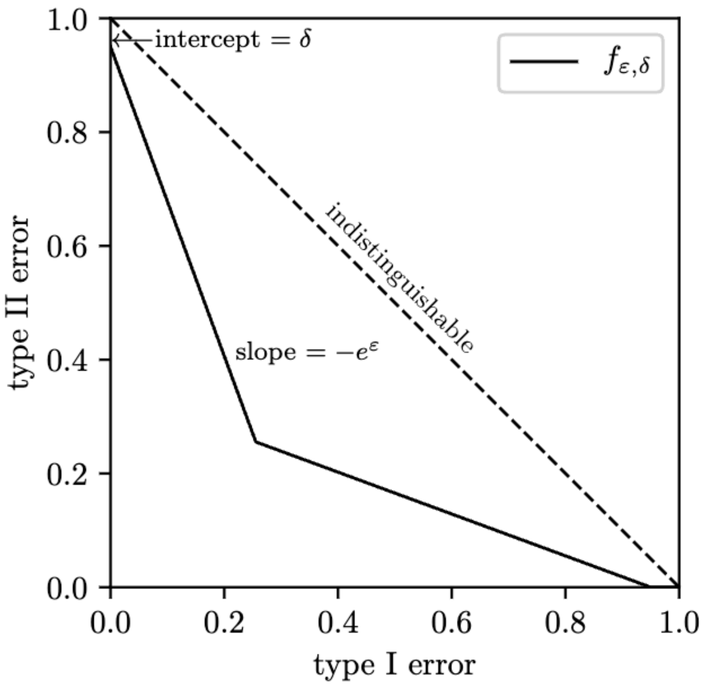
# Privacy Auditing

- Some decisions to make
  - Which $\tau$? Can try them all - will get a tradeoff curve.



# Privacy Auditing

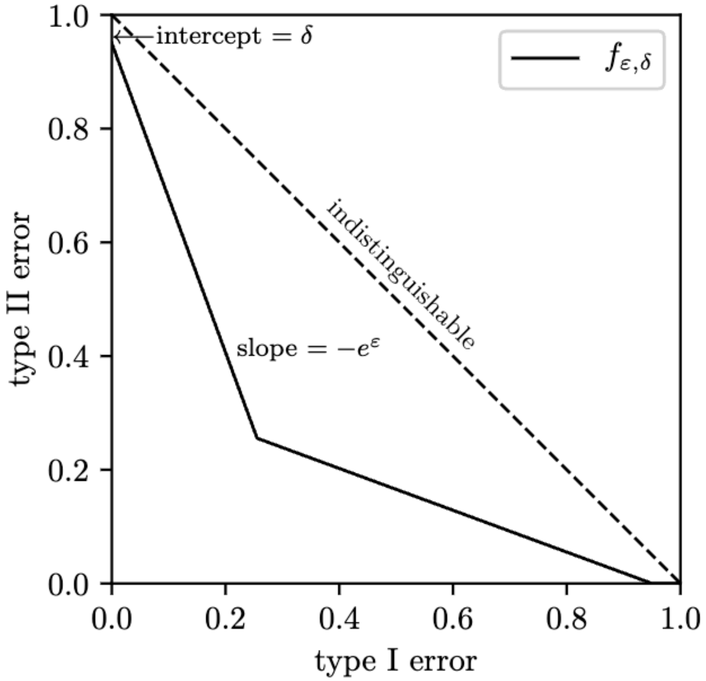
- Claimed privacy: (0.21, 10–5)-DP.
- With a threshold τ = 2.64 , attack had true positive rate of 4.922% and false positive rate of 0.174%.
- Is this possible?



# Aside: Clopper-Pearson “exact” method

- $Y = \frac{1}{n}\sum_{i=1}^{n} X_i$, where $X_i \sim$ Bern($\alpha$). $\alpha$ is unknown.
- Given Y for n observations, what can we say about $\alpha$?
- Clopper-Pearson gives intervals $\alpha \in [\alpha^-, \alpha^+]$ with probability $\geq 1 - p$
- No closed form - need to compute numerically.

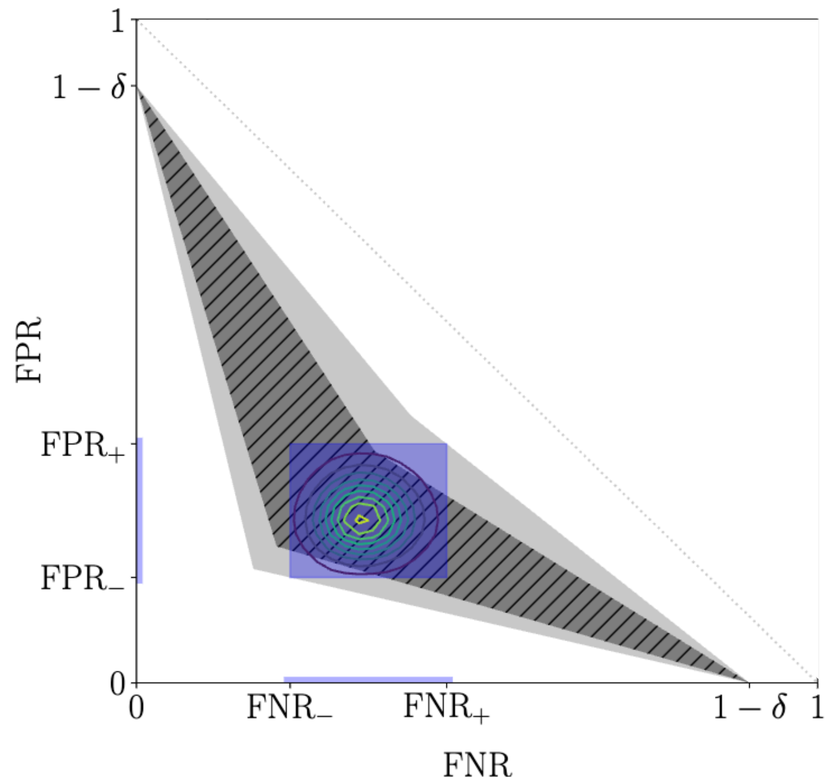
# Privacy Auditing




- We have claimed $\beta = 0.00174$ and $\alpha =$ 1- 4.922/100 = 0.95078.
- We have claimed privacy of (0.21, 10–5)-DP.
- $\beta \geq \max(1 - 10^{-5} - e^{0.21}0.95078\ ,\ (1 - 10^{-5} - 0.95078)/(e^{0.21})$ = 0.03988885074
- Can be due to sampling?
- By Clopper-Pearson, $\alpha^{+} \leq 0.95509$, $\beta^{-} \geq 0.00274$ with $p = 10^{-10}$
- Later, they found a bug and retracted the paper. Very common in DP!!

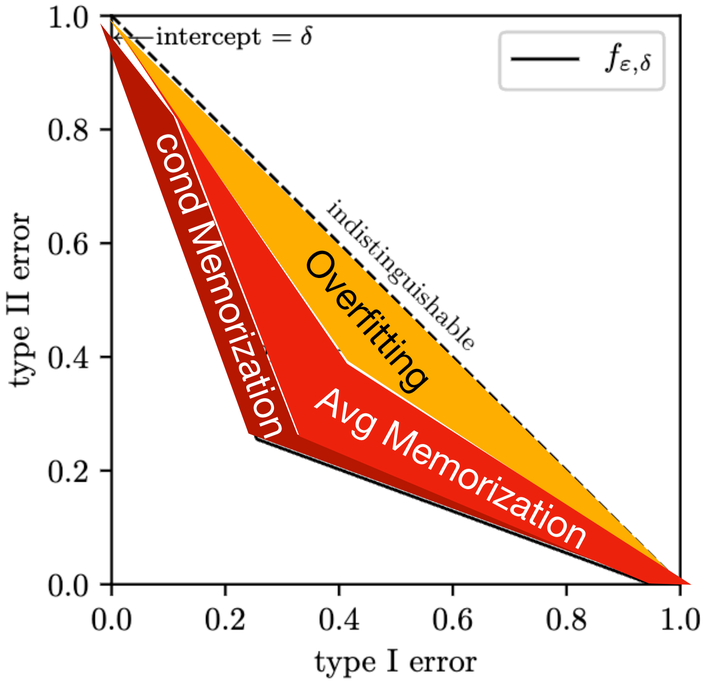
# Improvements: better stats

- Do we really need $\alpha^+, \beta^-$?
  - Directly bound $\log(\frac{1-\delta-\beta}{\alpha})$ using *Log-Katz confidence intervals*.
- Incorporate priors [ZB+23]:
  - Use Bayesian approach
  - Compute joint posterior of $\alpha, \beta, \varepsilon$
- Your favorite stats trick



# Improvements: picking images

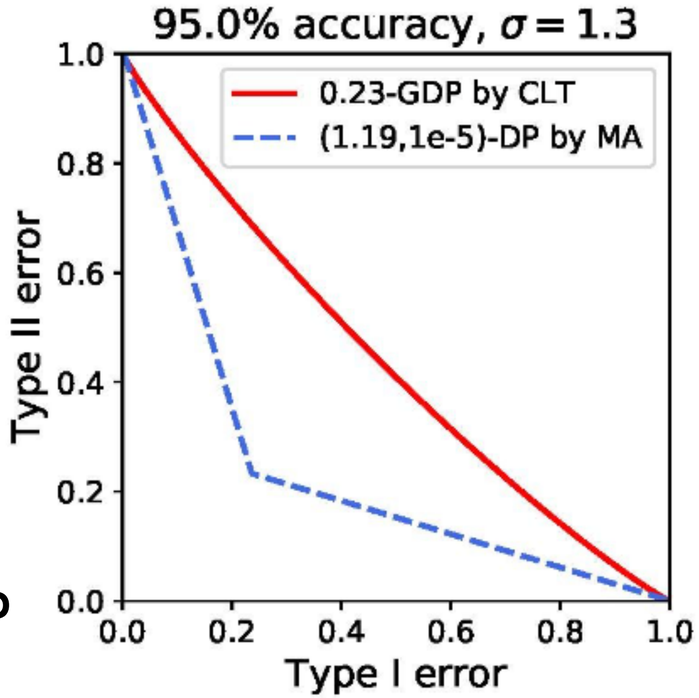
- Picking the right $(x', y')$ is an art
  - Very similar to backdoor attacks
- Goal is to test for *conditional memorization*
- Means searching for a “planted signal”
  - when detected, we are sure. i.e. low type I
  - but can miss a lot i.e. high type II
  - what if $\delta \geq \alpha$?

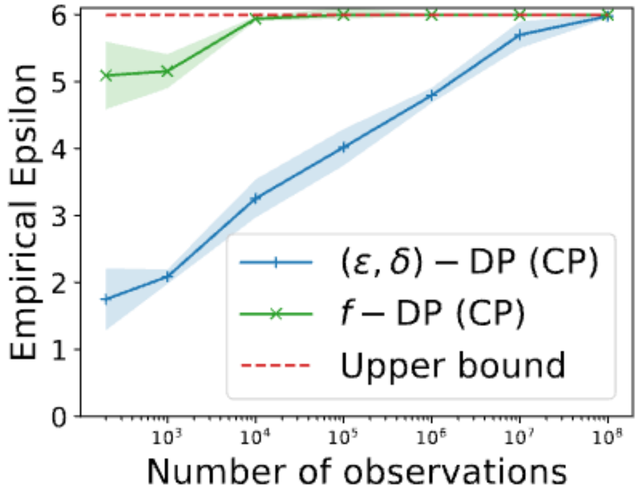


# Gaussian Membership Inference

## More improvements

- Test for GDP instead:
  - Suppose some Gaussian mechanism claims $(\varepsilon, \delta)$-DP
  - Calculate corresponding $\mu$-GDP
  - Check if empirical $\alpha, \beta$ allows such $\mu$
    $\mu^- = \Phi^{-1}(1 - \alpha^+) - \Phi^{-1}(\beta^-)$
  - Reduces number of runs by 10,000x [N+23]




(d) $\varepsilon = 6$

# Auditing with stronger adversaries

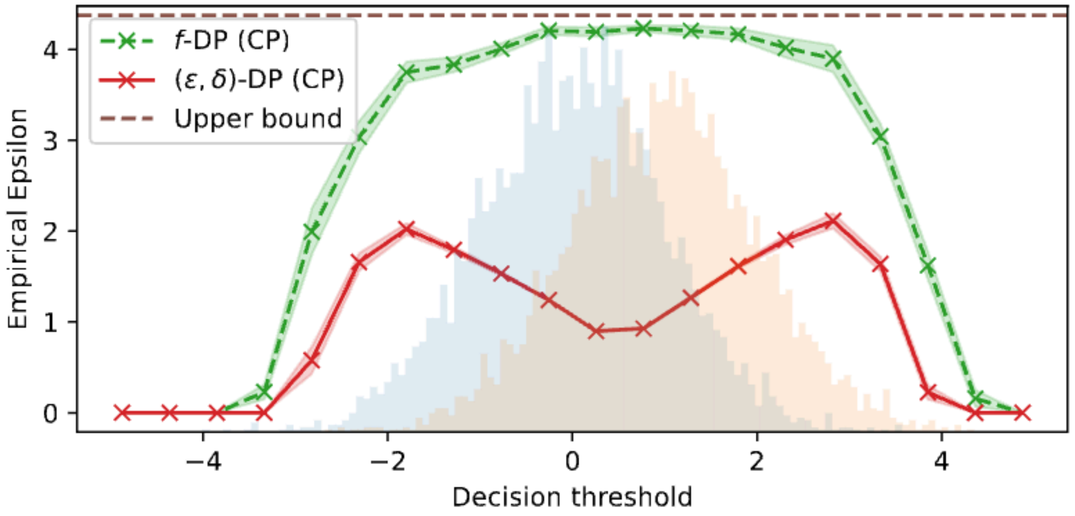
## Gradient canary

- We know we are running gradient descent. Mess with gradients directly
- At each time step t we will run 2 training runs in parallel:
    - Sample 2 batches i.i.d. with prob. $q$: $B_t$ and $B'_t$
    - Compute gradients
    - With prob $q$, add a canary gradient $g'$ to gradients of $B'_t$
    - Continue private training algorithm
    - Compare $O_t = \nabla_t^\top g'$ and $O'_t = \nabla_t'^\top g'$

# Auditing with stronger adversaries

## Gradient canary

- Compare $\nabla_t^\top g'$ and $\nabla_t'^\top g'$
- Sample $g'$ randomly - from Gaussian or Dirac
  - In high dimensions, random vectors are orthogonal i.e. we $\nabla_t^\top g' \approx 0$
  - True even after clipping and adding noise
  - But, $\nabla_t'^\top g' \approx \nabla_t^\top g' + q\|g'\|_2 \approx q\tau$
- Gives per-step estimate of $\varepsilon$.
  - Use composition to compute after $t$-rounds




- Questions: can we
  - simplify to use only a single batch?
  - Use the same $g'$ across t?

# Auditing with stronger adversaries

## Gradient canary

- Overview [N+23]:
  - Sample $g'$ from Dirac - random coordinate/ Gaussian
  - Estimate posterior distribution of $(\alpha, \beta)$ using Bayesian method [ZB+23]
  - Estimate per round $\varepsilon$ by comparing against sub-sampled Gaussian-DP
  - Combine with composition
- Can detect bugs in noise, clipping, etc. Cannot debug composition.

| Lower Bounding | Theoretical ε | CIFAR-10 WRN-16 |
|---|---|---|
| $f$-DP (CP) | 1 | 0.75 |
| | 4 | 3.40 |
| | 8 | 5.80 |
| | 16 | 11.14 |
| $f$-DP (ZB) | 1 | 0.95 |
| | 4 | 3.73 |
| | 8 | 7.09 |
| | 16 | 13.95 |
| $(\varepsilon, \delta)$-DP (CP) | 1 | 0.41 |
| | 4 | 1.37 |
| | 8 | 3.63 |
| | 16 | 5.25 |
| $(\varepsilon, \delta)$-DP (ZB) | 1 | 0.62 |
| | 4 | 2.65 |
| | 8 | 5.07 |
| | 16 | 5.25 |
| $\varepsilon-$DP (Katz) | 1 | 0.49 |
| | 4 | 1.65 |
| | 8 | 4.17 |
| | 16 | 7.52 |

# Auditing black-box models in a single run

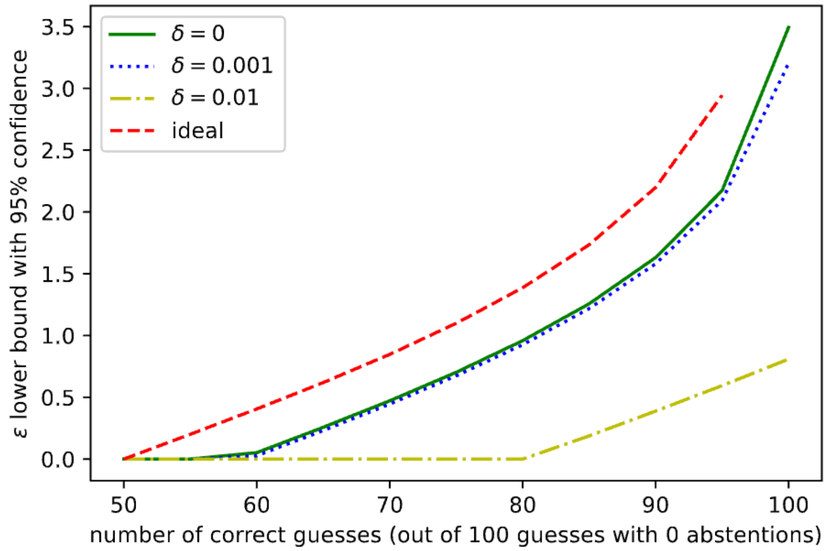
## Insert multiple canaries

- Gets even better if we insert multiple canaries.
- NeurIPS outstanding paper award! [SNJ23]
- Key idea: insert multiple canary datapoints
    - Include each of $m$ canaries randomly
    - Make m guesses - which canary was present?

# Auditing with stronger adversaries

## Multiple gradient canaries

- Select a set of canaries: $\mathcal{G} = \{g'_1, \ldots, g'_m\}$.
- For each $i \in [m]$, with prob. 0.5 include $g'_i \in \mathcal{G}'$. Otherwise it is dropped.
- At each time step t:
  - Sample datapoints with prob. $q$: batch $B_t$
  - With prob $q$, add each of the selected canaries $g'_i$ to gradients of $B_t$
  - Continue private training algorithm
  - Compute: $\{O_i = O_i + \nabla_t^\top g'_i\}$ for $i \in [m]$
- Sort the final $\{O_i\}$, declare top $m/2$ to have been included.
- Compare number of correct guesses with Theorem 5.2.



# Auditing black-box models in a single run

## Insert multiple canaries

- Key idea: insert multiple canary datapoints
    - Add $m$ canaries instead of just 1
    - Make m guesses - whether each canary was present
    - Works for black-box in single training run.
    - If white-box compare sums of dot-products.
- Very useful in practice: no need for composition
    - add canaries with some probability
    - try detect them if present
    - Compare number of correct guesses to compute $\varepsilon$ estimate.