CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Privacy I & II

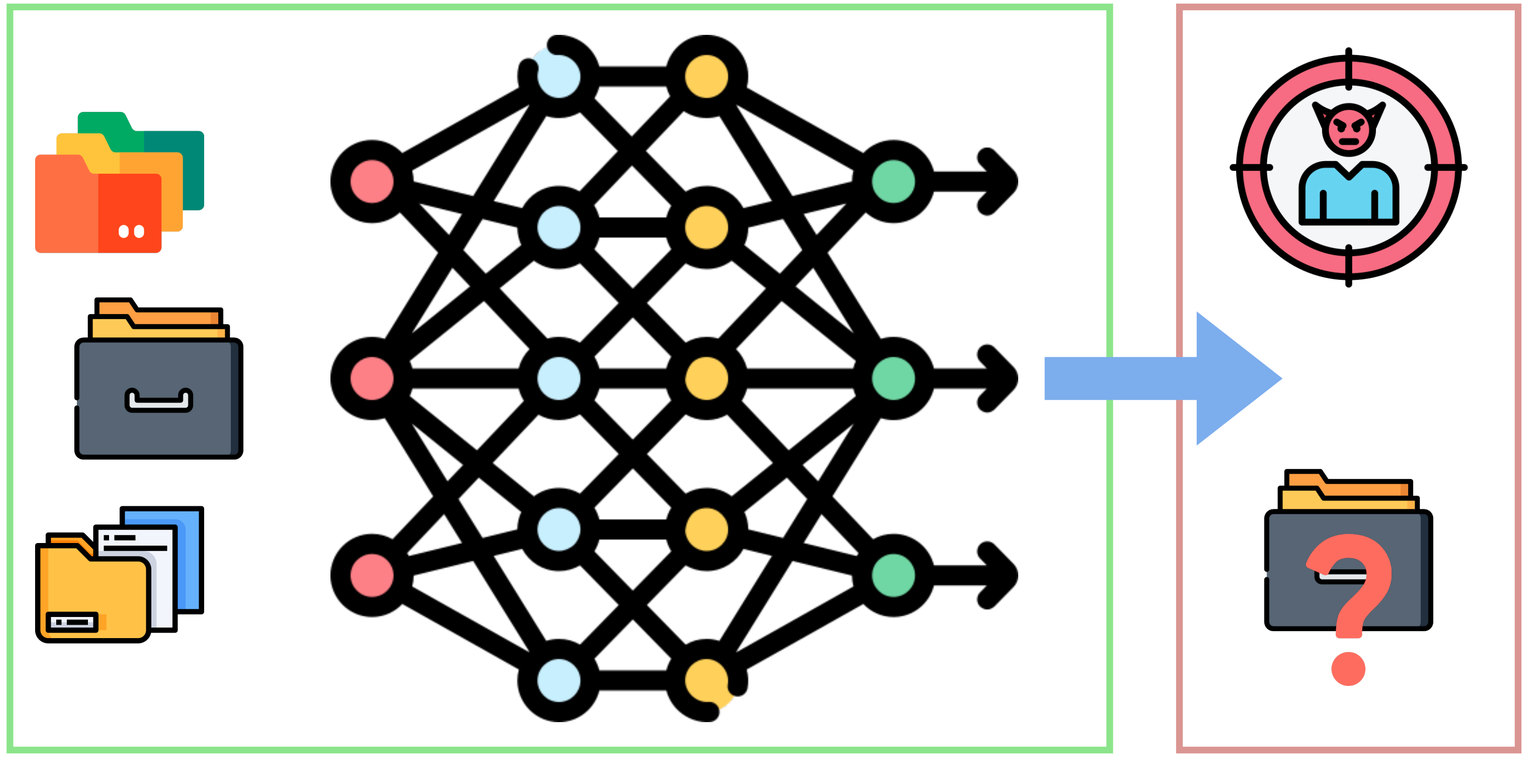
Tutorial Session 2 - Trustworthy Machine Learning

Adam Dziedzic & Franziska Boenisch
Nima Dindarsafa | 29.04.2026

Assignment 1 - Membership Inference Attack

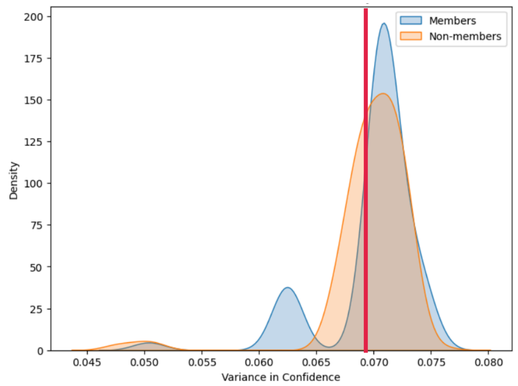
Questions?

Feedbacks?



Membership Inference Attack - Methods

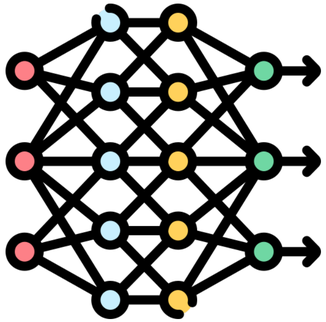
Threshold Based



Non-Member Member

Shadow-Model Based

Conf = (0.24)



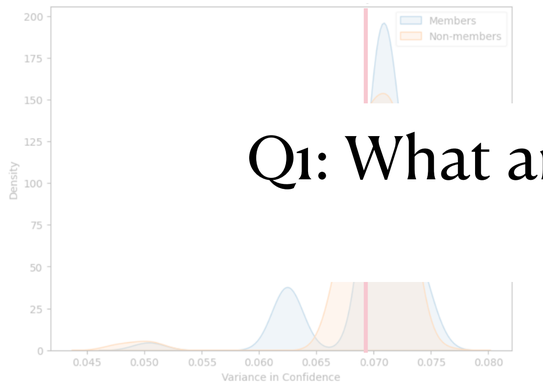
Out

Likelihood Ratio Attack

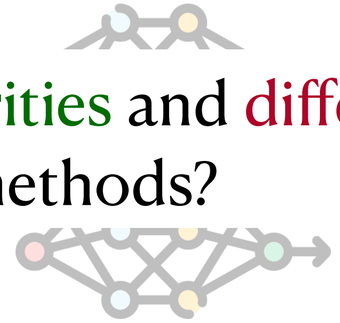
$$\Lambda = \frac{p(conf_{obs} \mid \mathcal{N}(\mu_{in}, \sigma_{in}^2))}{p(conf_{obs} \mid \mathcal{N}(\mu_{out}, \sigma_{out}^2))}$$

Membership Inference Attack - Methods

Threshold Based



Shadow-Model Based



Likelihood Ratio Attack

$$\frac{\mathcal{N}(\mu_{in}, \sigma_{in}^2))}{\mathcal{N}(\mu_{out}, \sigma_{out}^2))}$$

Q1: What are the core similarities and differences between these methods?

Membership Inference Attack - Methods

Detect member vs. non-member directly by thresholding the membership inference feature (e.g. Loss, Confidence etc.)

Main issue:

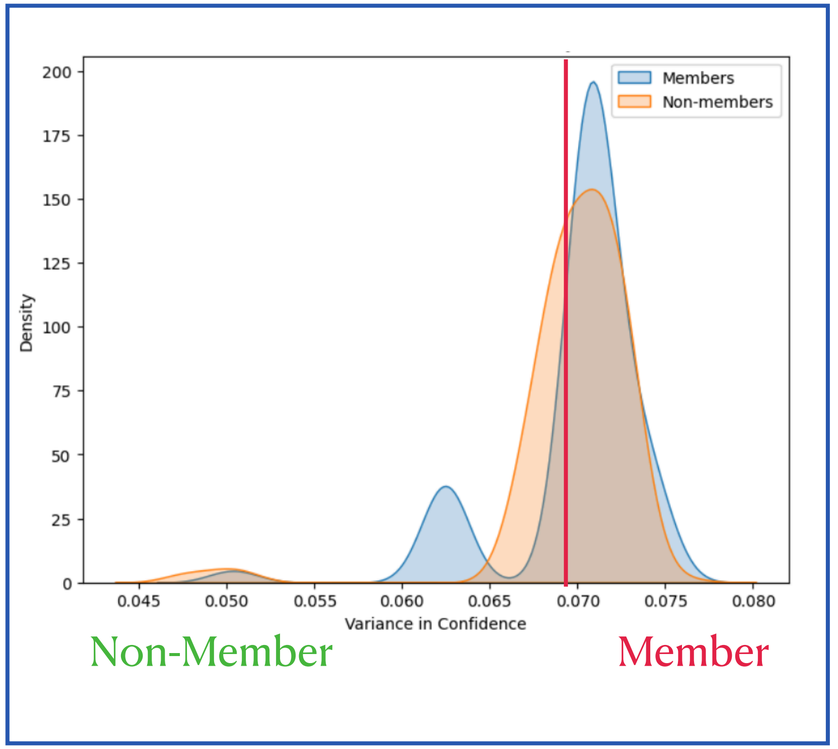
How can we **calibrate** this score?

Calibration: How well a model's predicted confidence scores correspond to the true probability of correctness.

Why we care?

- To avoid **False Positive** detection!

Threshold Based



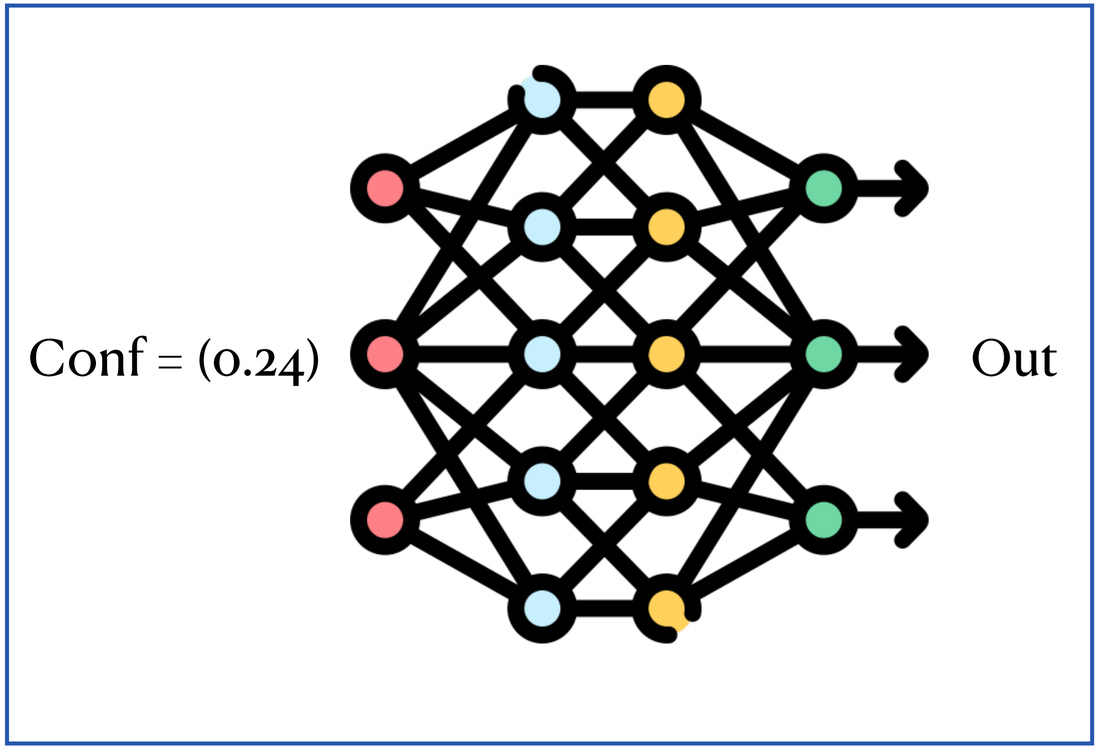
Membership Inference Attack - Methods

Process:

1. Use dataset sampled from the target model's training data distribution to train N **shadow models**.
2. Train half shadow models with the target data point and half without
3. Train a binary classifier using pairs of ("membership", "confidence")

The classifier **learns** to separate members from non-members.

Shadow-Model Based



Membership Inference Attack - Methods

Process:

1. Same shadow model training process.
2. But with a different objective.

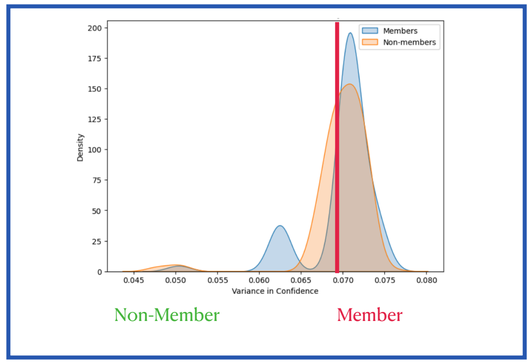
Do **statistical hypothesis testing** instead of classification!

Likelihood Ratio Attack

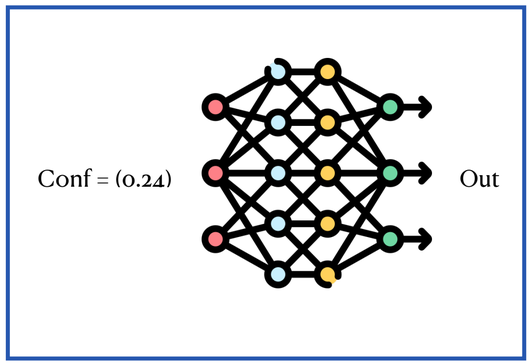
$$\Lambda = \frac{p(conf_{obs} \mid \mathcal{N}(\mu_{in}, \sigma_{in}^2))}{p(conf_{obs} \mid \mathcal{N}(\mu_{out}, \sigma_{out}^2))}$$

Membership Inference Attack - Methods

Threshold Based



Shadow-Model Based



Likelihood Ratio Attack

$$\Lambda = \frac{p(conf_{obs} \mid \mathcal{N}(\mu_{in}, \sigma_{in}^2))}{p(conf_{obs} \mid \mathcal{N}(\mu_{out}, \sigma_{out}^2))}$$

Similarities

All 3 methods use a membership inference features to draw conclusion about membership status.

Both shadow-model based and LiRA use shadow models to do this task.

But in different ways!

Differences

Shadow-model based method uses a classifier to **learn a discriminative decision boundary** between members and non-members.

LiRA, uses shadow models to **estimate the underlying probability distribution**.

Membership Inference Attack - Performance

Q2: Consider the following results for MIA

A. Threshold = 80%. Find TPR, FPR and Accuracy

B. Calculate TPR@20%-FPR

C. Why TPR@low-FPR not AUC or Accuracy?

ID	True Status	Confidence	Attack Prediction
1	Member	0.96	Member
2	Member	0.91	Member
3	Member	0.83	Member
4	Member	0.74	Non-member
5	Member	0.62	Non-member
6	Non-member	0.88	Member
7	Non-member	0.77	Non-member
8	Non-member	0.69	Non-member
9	Non-member	0.55	Non-member
10	Non-member	0.40	Non-member

Membership Inference Attack - Performance

Part A.

Threshold = 80% → Predict member if confidence ≥ 0.80

	Predicted Member	Predicted Non-member
True Member	TP = 3	FN = 2
True Non-member	FP = 1	TN = 4

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{5} = 60\,\%$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{1}{5} = 20\,\%$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total}} = \frac{7}{10} = 70\,\%$$

True Positive

False Negative

False Positive

True Negative

ID	True Status	Confidence	Attack Prediction
1	Member	0.96	Member
2	Member	0.91	Member
3	Member	0.83	Member
4	Member	0.74	Non-member
5	Member	0.62	Non-member
6	Non-member	0.88	Member
7	Non-member	0.77	Non-member
8	Non-member	0.69	Non-member
9	Non-member	0.55	Non-member
10	Non-member	0.40	Non-member

Membership Inference Attack - Performance

Part B.

Already solved!

TPR@20%-FPR = Adjust the threshold such that only 20% of non-members are **falsely accused**.

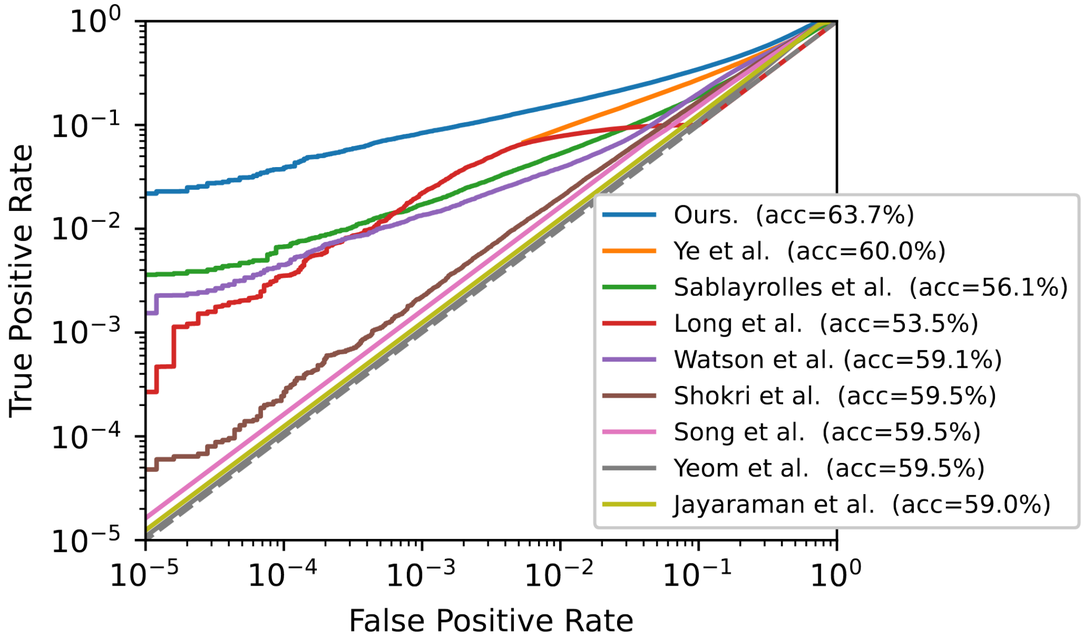
20% of 10 observations is equivalent to only 1 data point in our problem.

ID	True Status	Confidence	Attack Prediction
1	Member	0.96	Member
2	Member	0.91	Member
3	Member	0.83	Member
4	Member	0.74	Non-member
5	Member	0.62	Non-member
6	Non-member	0.88	Member
7	Non-member	0.77	Non-member
8	Non-member	0.69	Non-member
9	Non-member	0.55	Non-member
10	Non-member	0.40	Non-member

Membership Inference Attack - Performance

Part C.

- In privacy, false accusations (FPR) is costly!
- Accuracy or AUC can look good on average while hiding poor performance in the low FPR region.
- Privacy attack goal: The attacker can identify true members while making very few false accusations.



ROC Curve. Source [1]

[1] Membership Inference Attacks From First Principles; Carlini et al. 2021

Membership Inference Attack - LiRA

Classification

Learn a decision boundary between confidence of members vs. non-members.

Limited to the shadow dataset and shadow models.

Highly likely to overfit to the data.

Hypothesis Testing

Perform a statistical hypothesis test.

Estimate how **likely** it is to observe such an outcome given the estimated underlying probability distribution.

$$\text{Likelihood Ratio} = \frac{P(\text{observing confidence x given in distribution})}{P(\text{observing confidence x given out distribution})}$$

Also due to **Neyman-Pearson lemma**!

Membership Inference Attack - LiRA

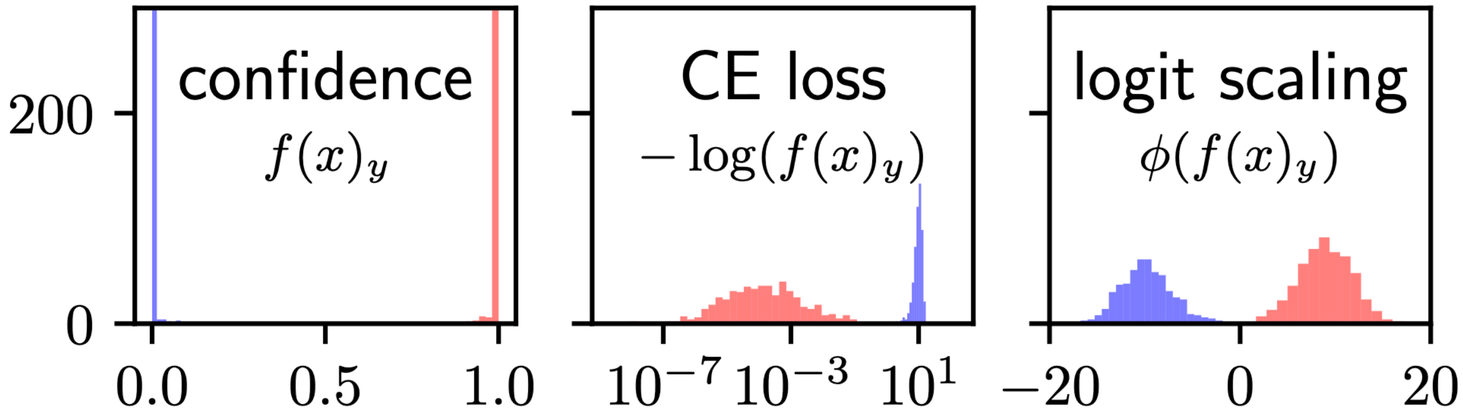


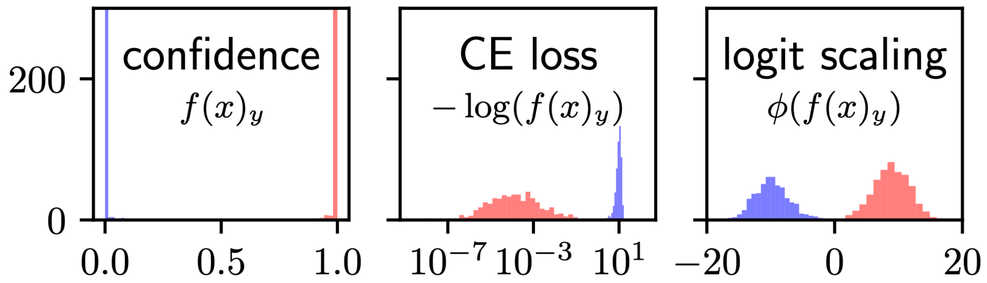
Figure Source: [1]

Q3- part (b): Why is logit scaling applied on model confidence?

- Model confidence is bounded in [0,1] interval and thus **not normally distributed**!
- Logit scaling model confidence maps it to $(-\infty, \infty)$ and **empirically** normally distributed.

[1] Membership Inference Attacks From First Principles; Carlini et al. 2021

Membership Inference Attack - LiRA



Q3- part (c): LiRA assumptions

$$\Lambda = \frac{p(conf_{obs} \mid \mathcal{N}(\mu_{in}, \sigma_{in}^2))}{p(conf_{obs} \mid \mathcal{N}(\mu_{out}, \sigma_{out}^2))}$$

Implicit assumption is that logit of model confidence is **normally distributed**.

If violated:

- Use non-parametric density estimation, such as kernel density estimation (KDE)
- Consequence? Lower **power**!

Membership Inference Attack - RMIA

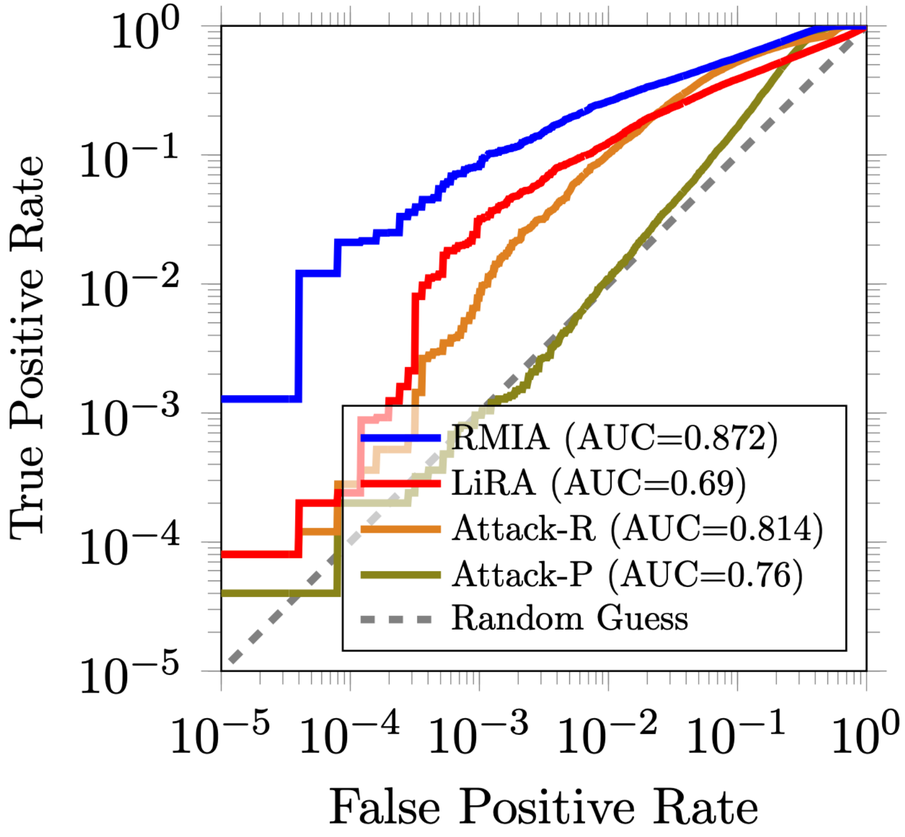
Null Hypothesis H_0: The composition of worlds in which the target data point x is replaced with a random data point z sampled from the population.

Pairwise Likelihood Ratio Tests: Testing the membership of a data point x relative to another data point z.

$$\mathrm{LR}_\theta = \frac{\Pr(\theta \mid x)}{\Pr(\theta \mid z)}$$

Decision rule:

$$\mathrm{Score}_{\mathrm{MIA}}(x;\theta) = \Pr_{z\sim\pi}\left(\mathrm{LR}_\theta(x,z) \geq \gamma\right)$$



Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

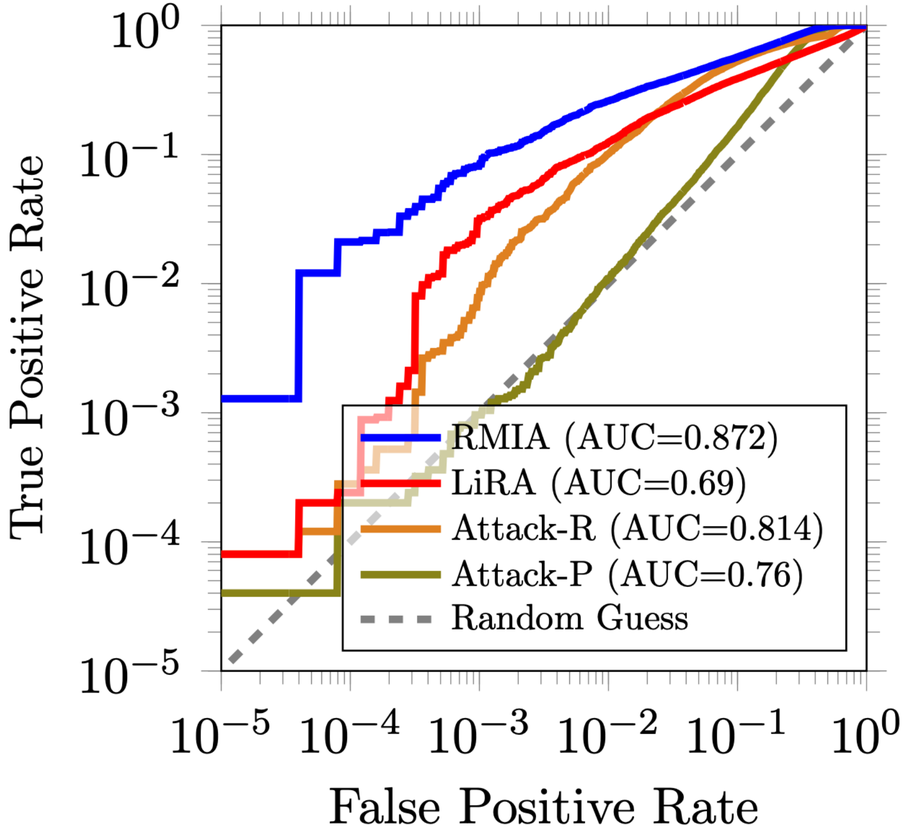
How to compute the pairwise likelihood ratios?

$$\mathrm{LR}_\theta = \frac{\Pr(\theta \mid x)}{\Pr(\theta \mid z)}$$

Use Bayes rule!

$$\mathrm{LR}_\theta = \left(\frac{\Pr(x \mid \theta)}{\Pr(x)}\right)\left(\frac{\Pr(z \mid \theta)}{\Pr(z)}\right)^{-1}$$

$\Pr(x \mid \theta)$ = Prediction score (soft-max) of output of the model $f_\theta(x)$



Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

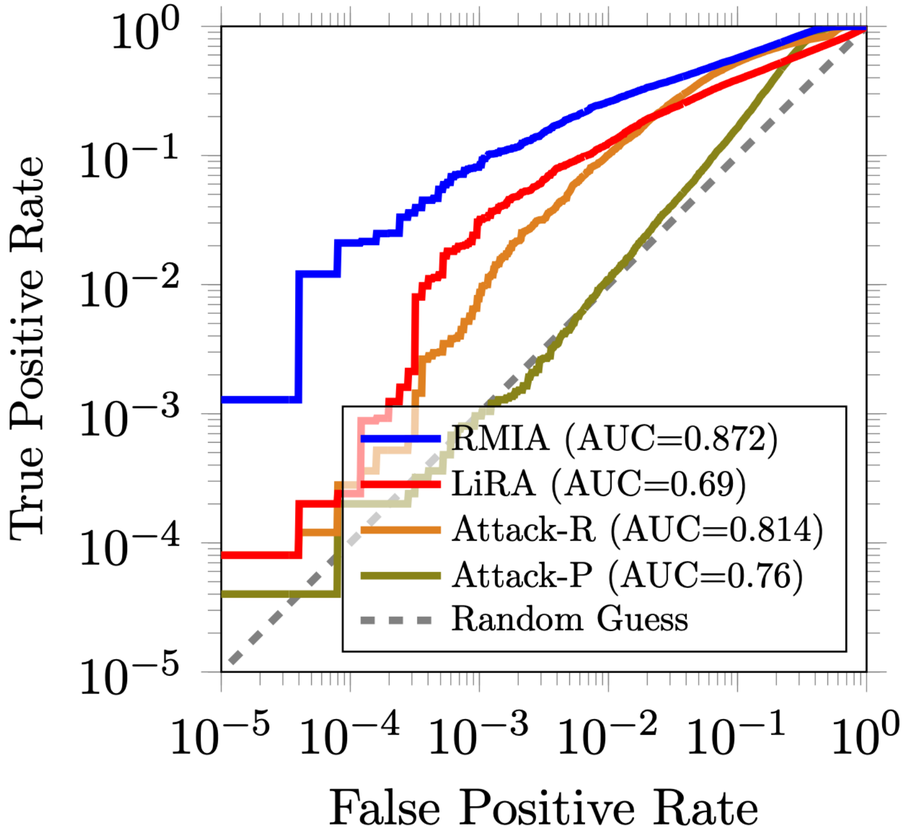
What about normalizing constant $\Pr(x)$?

$$\Pr(x) = \sum_{\theta'} \Pr(x \mid \theta')$$

$$= \sum_{D,\theta'} \Pr(x \mid \theta') \Pr(\theta' \mid D) \Pr(D)$$

However, in practice

$$\Pr(x) \approx \frac{1}{k} \sum_{j=1}^{k} \Pr(x \mid \theta'_j)$$



Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

Algorithm 1 MIA Score Computation with RMIA. The input to this algorithm is k reference models Θ, the target model θ, target (test) sample x, parameter γ, and a scaling factor a as described in Appendix B.2.2. We assume the reference models Θ are pre-trained on random samples from a population dataset available to adversary; each sample from the population dataset is included in training of half of reference models. The algorithm also takes an *online* flag which indicates whether we intend to run MIA in the online mode.

Randomly choose a subset Z from the population dataset
$C \leftarrow 0$
if *online* **then**
  $\Theta_{in} \leftarrow \emptyset$
  for k times **do**
    $D_i \leftarrow$ randomly sample a dataset from population data π
    $\theta_x \leftarrow \mathcal{T}(D_i \cup x)$
    $\Theta_{in} \leftarrow \Theta_{in} \cup \{\theta_x\}$
  end for
  $\Pr(x) \leftarrow \frac{1}{2k}\left(\sum_{\theta' \in \Theta} \Pr(x|\theta') + \sum_{\theta' \in \Theta_{in}} \Pr(x|\theta')\right)$ (See equation 9)
else
  $\Pr(x)_{OUT} \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(x|\theta')$
  $\Pr(x) \leftarrow \frac{1}{2}\left((1+a).\Pr(x)_{OUT} + (1-a)\right)$ (See Appendix B.2.2)
end if
$Ratio_x \leftarrow \frac{\Pr(x|\theta)}{\Pr(x)}$
for each sample z in Z **do**
  $\Pr(z) \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(z|\theta')$
  $Ratio_z \leftarrow \frac{\Pr(z|\theta)}{\Pr(z)}$
  if $(Ratio_x / Ratio_z) > \gamma$ **then**
    $C \leftarrow C + 1$
  end if
end for
$\text{Score}_{\text{MIA}}(x;\theta) \leftarrow C/|Z|$ (See equation 5)

Note that RMIA has 2 versions:
Online and Offline

Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

Algorithm 1 MIA Score Computation with RMIA. The input to this algorithm is k reference models Θ, the target model θ, target (test) sample x, parameter γ, and a scaling factor a as described in Appendix B.2.2. We assume the reference models Θ are pre-trained on random samples from a population dataset available to adversary; each sample from the population dataset is included in training of half of reference models. The algorithm also takes an *online* flag which indicates whether we intend to run MIA in the online mode.

Randomly choose a subset Z from the population dataset
$C \leftarrow 0$
if *online* **then**
  $\Theta_{in} \leftarrow \emptyset$
  for k times **do**
    $D_i \leftarrow$ randomly sample a dataset from population data π
    $\theta_x \leftarrow \mathcal{T}(D_i \cup x)$
    $\Theta_{in} \leftarrow \Theta_{in} \cup \{\theta_x\}$
  end for
  $\Pr(x) \leftarrow \frac{1}{2k}\left(\sum_{\theta' \in \Theta} \Pr(x|\theta') + \sum_{\theta' \in \Theta_{in}} \Pr(x|\theta')\right)$ (See equation 9)
else
  $\Pr(x)_{OUT} \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(x|\theta')$
  $\Pr(x) \leftarrow \frac{1}{2}\left((1+a).\Pr(x)_{OUT} + (1-a)\right)$ (See Appendix B.2.2)
end if
$Ratio_x \leftarrow \frac{\Pr(x|\theta)}{\Pr(x)}$
for each sample z in Z **do**
  $\Pr(z) \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(z|\theta')$
  $Ratio_z \leftarrow \frac{\Pr(z|\theta)}{\Pr(z)}$
  if $(Ratio_x / Ratio_z) > \gamma$ **then**
    $C \leftarrow C + 1$
  end if
end for
$\text{Score}_{\text{MIA}}(x;\theta) \leftarrow C/|Z|$ (See equation 5)

For the Online version, we train k IN shadow models where the data point x is **included**.

Using the IN shadow models estimate $\Pr(x)$.

Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

Algorithm 1 MIA Score Computation with RMIA. The input to this algorithm is k reference models Θ, the target model θ, target (test) sample x, parameter γ, and a scaling factor a as described in Appendix B.2.2. We assume the reference models Θ are pre-trained on random samples from a population dataset available to adversary; each sample from the population dataset is included in training of half of reference models. The algorithm also takes an *online* flag which indicates whether we intend to run MIA in the online mode.

Randomly choose a subset Z from the population dataset
$C \leftarrow 0$
if *online* **then**
  $\Theta_{in} \leftarrow \emptyset$
  for k times **do**
    $D_i \leftarrow$ randomly sample a dataset from population data π
    $\theta_x \leftarrow \mathcal{T}(D_i \cup x)$
    $\Theta_{in} \leftarrow \Theta_{in} \cup \{\theta_x\}$
  end for
  $\Pr(x) \leftarrow \frac{1}{2k}\left(\sum_{\theta' \in \Theta} \Pr(x|\theta') + \sum_{\theta' \in \Theta_{in}} \Pr(x|\theta')\right)$ (See equation 9)
else
  $\Pr(x)_{OUT} \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(x|\theta')$
  $\Pr(x) \leftarrow \frac{1}{2}\left((1+a).\Pr(x)_{OUT} + (1-a)\right)$ (See Appendix B.2.2)
end if
$Ratio_x \leftarrow \frac{\Pr(x|\theta)}{\Pr(x)}$
for each sample z in Z **do**
  $\Pr(z) \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(z|\theta')$
  $Ratio_z \leftarrow \frac{\Pr(z|\theta)}{\Pr(z)}$
  if $(Ratio_x / Ratio_z) > \gamma$ **then**
    $C \leftarrow C + 1$
  end if
end for
$\text{Score}_{\text{MIA}}(x;\theta) \leftarrow C/|Z|$ (See equation 5)

The Offline version only uses the OUT shadow models to draw the conclusion.

Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Membership Inference Attack - RMIA

Algorithm 1 MIA Score Computation with RMIA. The input to this algorithm is k reference models Θ, the target model θ, target (test) sample x, parameter γ, and a scaling factor a as described in Appendix B.2.2. We assume the reference models Θ are pre-trained on random samples from a population dataset available to adversary; each sample from the population dataset is included in training of half of reference models. The algorithm also takes an *online* flag which indicates whether we intend to run MIA in the online mode.

Randomly choose a subset Z from the population dataset
$C \leftarrow 0$
if *online* **then**
  $\Theta_{in} \leftarrow \emptyset$
  for k times **do**
    $D_i \leftarrow$ randomly sample a dataset from population data π
    $\theta_x \leftarrow \mathcal{T}(D_i \cup x)$
    $\Theta_{in} \leftarrow \Theta_{in} \cup \{\theta_x\}$
  end for
  $\Pr(x) \leftarrow \frac{1}{2k}\left(\sum_{\theta' \in \Theta} \Pr(x|\theta') + \sum_{\theta' \in \Theta_{in}} \Pr(x|\theta')\right)$ (See equation 9)
else
  $\Pr(x)_{OUT} \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(x|\theta')$
  $\Pr(x) \leftarrow \frac{1}{2}\left((1+a).\Pr(x)_{OUT} + (1-a)\right)$ (See Appendix B.2.2)
end if
$Ratio_x \leftarrow \frac{\Pr(x|\theta)}{\Pr(x)}$
for each sample z in Z **do**
  $\Pr(z) \leftarrow \frac{1}{k}\sum_{\theta' \in \Theta} \Pr(z|\theta')$
  $Ratio_z \leftarrow \frac{\Pr(z|\theta)}{\Pr(z)}$
  if $(Ratio_x / Ratio_z) > \gamma$ **then**
    $C \leftarrow C + 1$
  end if
end for
$\text{Score}_{\text{MIA}}(x;\theta) \leftarrow C/|Z|$ (See equation 5)

Compute the likelihood ratio of random samples and compare it to that of the sample x. If it has passed the threshold γ then the memorization signal is larger than ordinary samples.

Reference: Low-Cost High-Power Membership Inference Attacks; Zarifzadeh et al.

Differential Privacy

A randomized algorithm M is said to be (ε, δ)-DP if the algorithm's outcome is "roughly the same" for two neighboring datasets d and d'.

An observer seeing the output of the algorithm cannot tell whether a **particular individual**'s information was used in the computation.

$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr[M(d') \in S] + \delta$$

Differential Privacy - Laplace Mechanism

Q1- part (a): Calculate the sensitivity for d

d = {(Alice ,0) ,(Bob, 1), (Carol, 0), (Dan, 1), (Eve, 0)}

f(d) = number of smokers in d = (current count) 2

Let's add (Nima, 1) to this dataset.

f(d') = number of smokers in d' = 3

$$\Delta f = \max(\left| f(d) - f(d') \right|) = 3 - 2 = 1$$

Differential Privacy - Laplace Mechanism

Q1- part (b): We want (0.5)-DP with Laplace mechanism. Solve for b.

$$\text{Lap}(x \mid b, \mu) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

We can choose $b = \frac{\Delta f}{\varepsilon}$ with $\epsilon = 0.5$ then $b = \frac{\Delta f}{\varepsilon} = \frac{1}{0.5} = 2$

$$M(d) = f(d) + Lap(0,2)$$

Achieves (0.5)-DP guarantee.

(Proof comes next)

Differential Privacy - Laplace Mechanism

Q1- part (b): Prove it!

Let's start by the ratio of probability of observing outcome z from algorithm M for two neighboring datasets:

$$\frac{p(M(d) = z)}{p(M(d') = z)} = \frac{\frac{1}{2b}\exp\left(-\frac{|z - f(d)|}{b}\right)}{\frac{1}{2b}\exp\left(-\frac{|z - f(d')|}{b}\right)}$$

With cancelling the constants we end up with the following expression:

$$= \exp\left(\frac{|z - f(d')| - |z - f(d)|}{b}\right)$$

By triangle inequality we have:

$$|z - f(d')| - |z - f(d)| \leq |f(d) - f(d')|$$

Differential Privacy - Laplace Mechanism

Q1- part (b): Prove it!

So we end up with the following inequality:

$$\frac{p(M(d) = z)}{p(M(d') = z)} \leq \exp\left(\frac{|f(d) - f(d')|}{b}\right)$$

Also, by the definition of sensitivity:

$$|f(d) - f(d')| \leq \Delta f$$

As a result:

$$\frac{p(M(d) = z)}{p(M(d') = z)} \leq \exp\left(\frac{\Delta f}{b}\right)$$

Substitute:

$$b = \frac{\Delta f}{\varepsilon}$$

Differential Privacy - Laplace Mechanism

Q1- part (b): Prove it!

The result would be:

$$\frac{p(M(d) = z)}{p(M(d') = z)} \leq e^{\varepsilon}$$

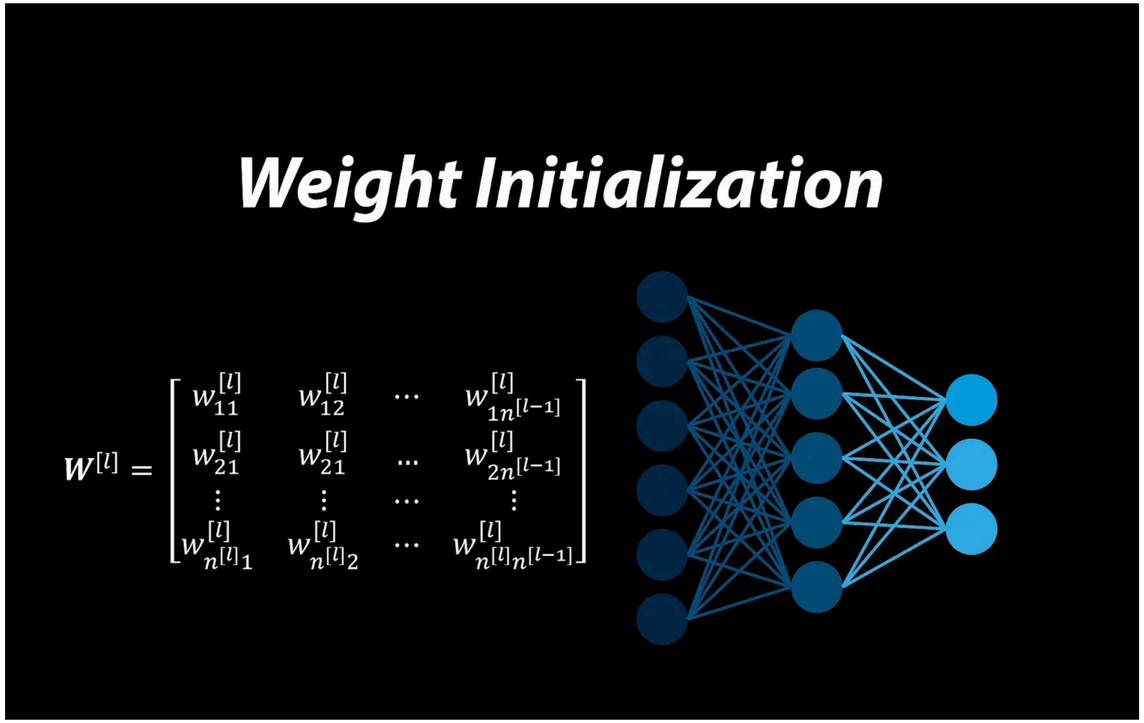
$$p(M(d) = z) \leq e^{\varepsilon} p(M(d') = z)$$

This is the definition of $\varepsilon -$ DP!

Differential Privacy - DP-SGD

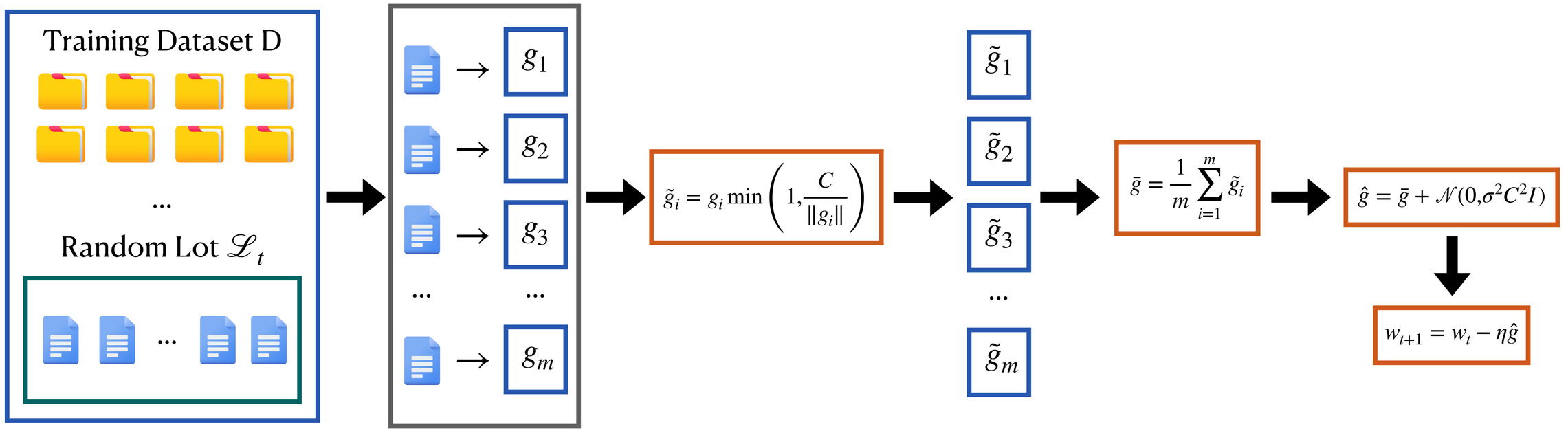
Question: Why can't Differential Privacy directly applied to the model weights?

In other words, measure the sensitivity on model weights and directly apply noise there.



Differential Privacy - DP-SGD

1. Sample random **lot**
2. Compute **per-example** gradient
3. **Clip** the gradients
4. Aggregate the gradients
5. Add **gaussian noise**



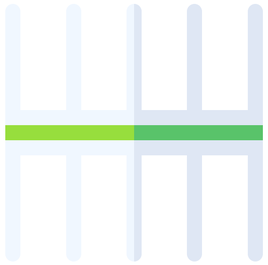
6. Update the model parameters

Differential Privacy - DP-SGD

Question 2 - part (a)

Batch

- Sampled **uniformly**
- Has a **fixed size** B
- Sampling done without replacement (often)



Lot

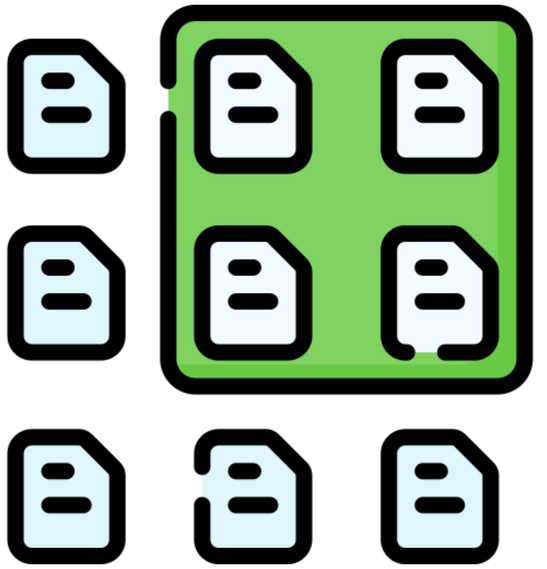
- Sampled with **Poisson subsampling**
- Each sample is included **independently** with probability $q = \frac{L}{N}$
- The lot size is a **random variable** (not fixed)

Differential Privacy - DP-SGD

Question 2 - part (a)

Privacy Amplification by Poisson Subsampling

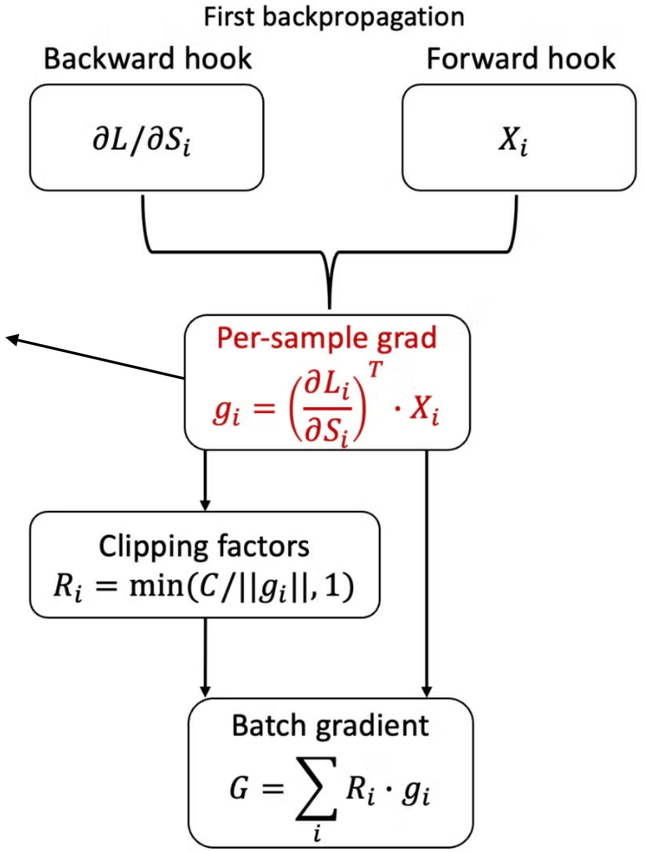
- Each data point is sampled **independently**
- So, in some runs the data is included and in other runs it is not
- The influence of the data point becomes diluted
- **Cumulative privacy leakage** is reduced.



Differential Privacy - DP-SGD

Per-example gradient

Full per-sample gradient matrix is formed here!



Ghost clipping

Ghost clipping requires second back propagation to calculate the batch gradient.

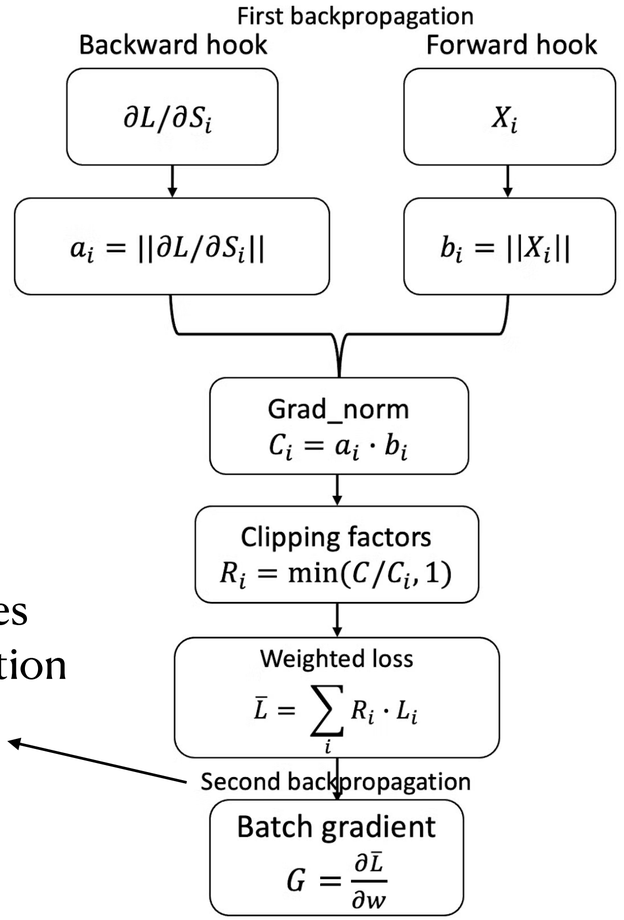


Image source: https://pytorch.org/blog/clipping-in-opacus/

Differential Privacy - PATE

Confidence Noisy Gaussian Max Aggregation

Input: Data point x, Threshold T, Noise parameters σ_1, σ_2

if $\max_i \{n_i(x) + \mathcal{N}(0, \sigma_1^2)\} \geq T$ then,

    return $\arg\max_j \{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$

else

    return $\perp$

Differential Privacy - PATE

Confidence Noisy Gaussian Max Aggregation

Input:

$n(x)$ = [42, 31, 15, 8, 4], T = 45

σ_1 = 4.7 , σ_2 = [-3.1, 6.8, 1.2, 9.5, 0.4]

Step 1: Check $\max_i \{n_i(x) + \mathcal{N}(0,\sigma_1^2)\} \geq T \rightarrow \max_i \{46.7,35.7,19.7,12.7,8.7\} \geq 45$

The if condition is True. Then we go to the second step.

Step 2: return $\arg\max_j \{n_j(x) + \mathcal{N}(0,\sigma_2^2)\} \rightarrow \arg\max_j \{38.9,37.8,16.2,17.5,4.4\} = 1$

Output: Return class 1 (considering that index starts from 1)

Questions?