CS5562: Trustworthy Machine Learning

Part II Lecture 3: Quantitative Reasoning About Data Privacy

Reza Shokri[a]

Aug 2023

[a]Acknowledgment. The wonderful teaching assistants: Hongyan Chang, Martin Strobel, Jiashu Tao, Yao Tong, Jiayuan Ye

Quantify privacy loss using Membership Inference Attacks

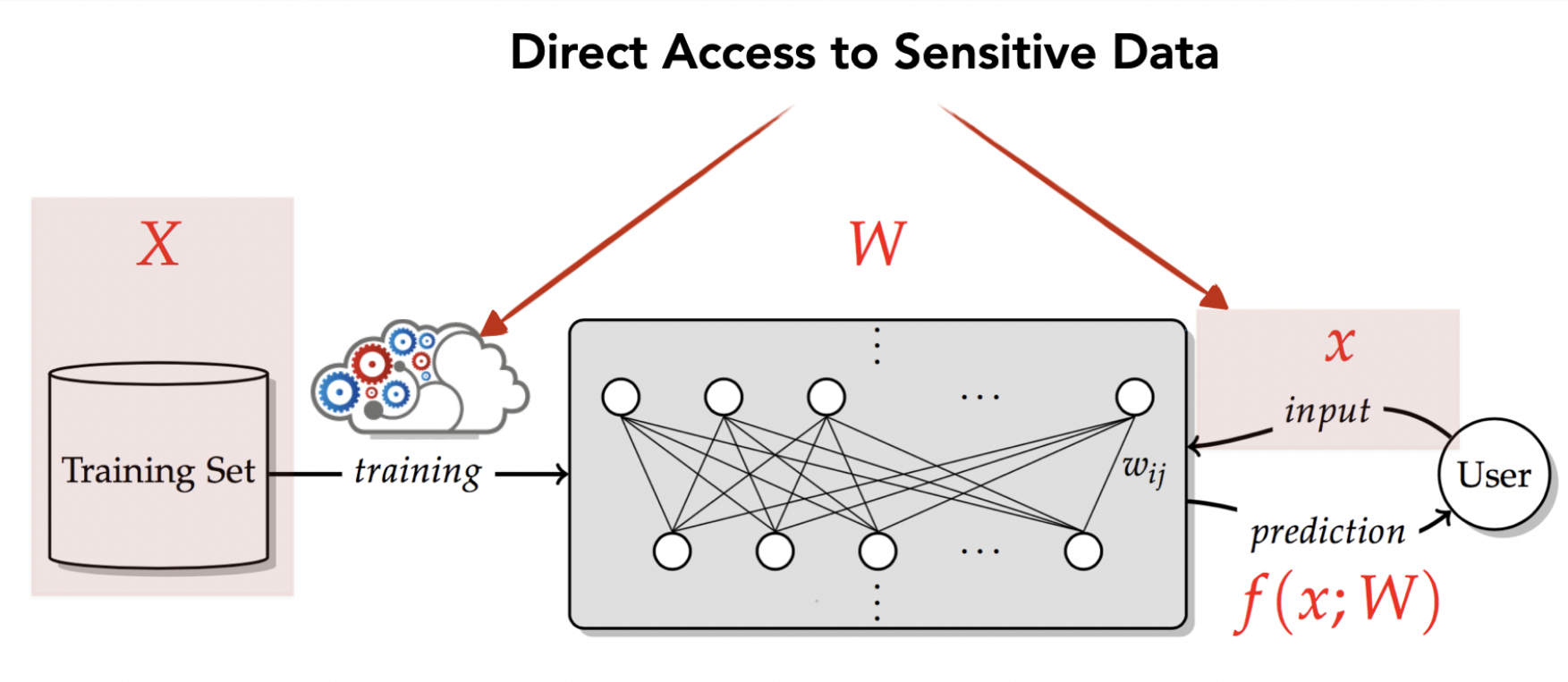
How to design powerful inference attacks

How to guarantee privacy (differentially private algorithms)

Privacy Regulations

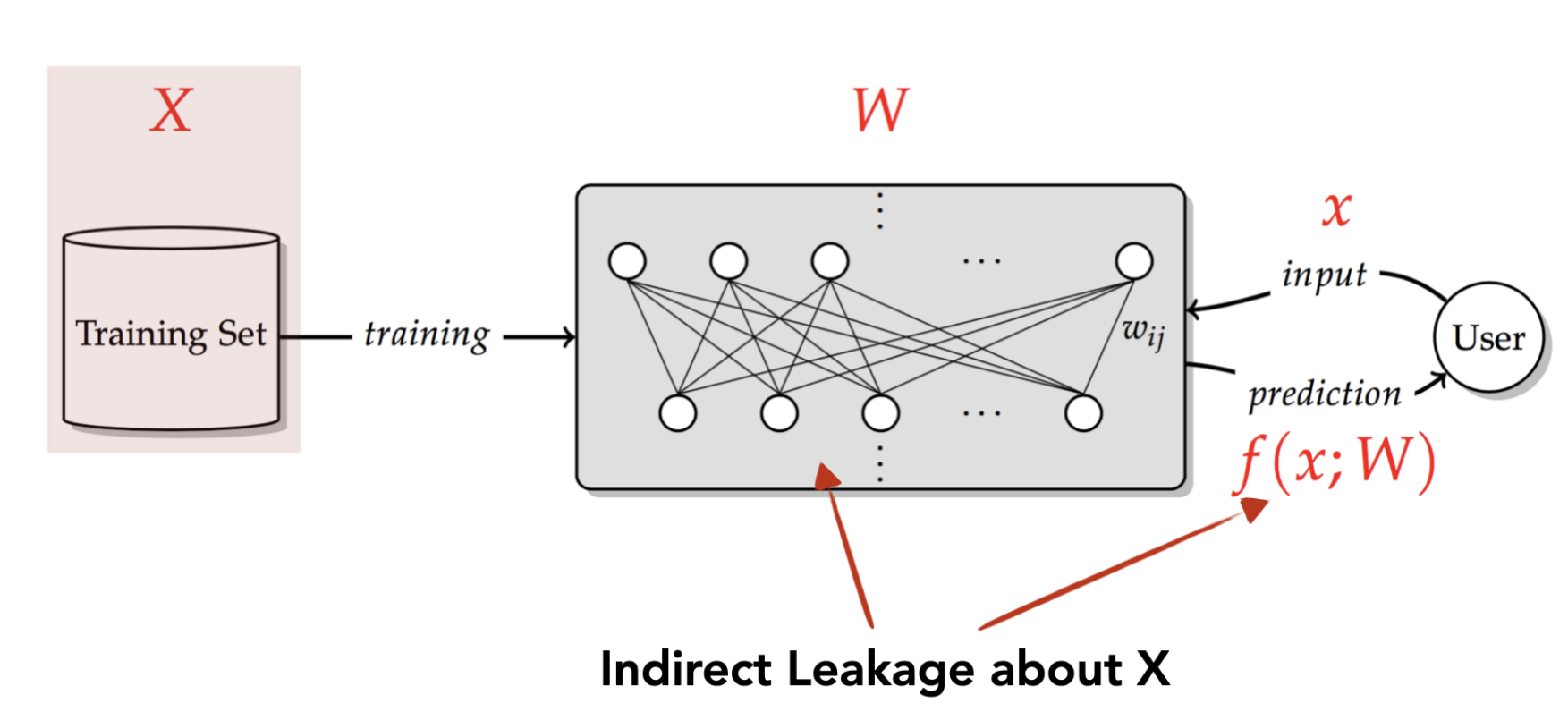
GDPR – Data Protection Impact Assessment

Direct Privacy Risks



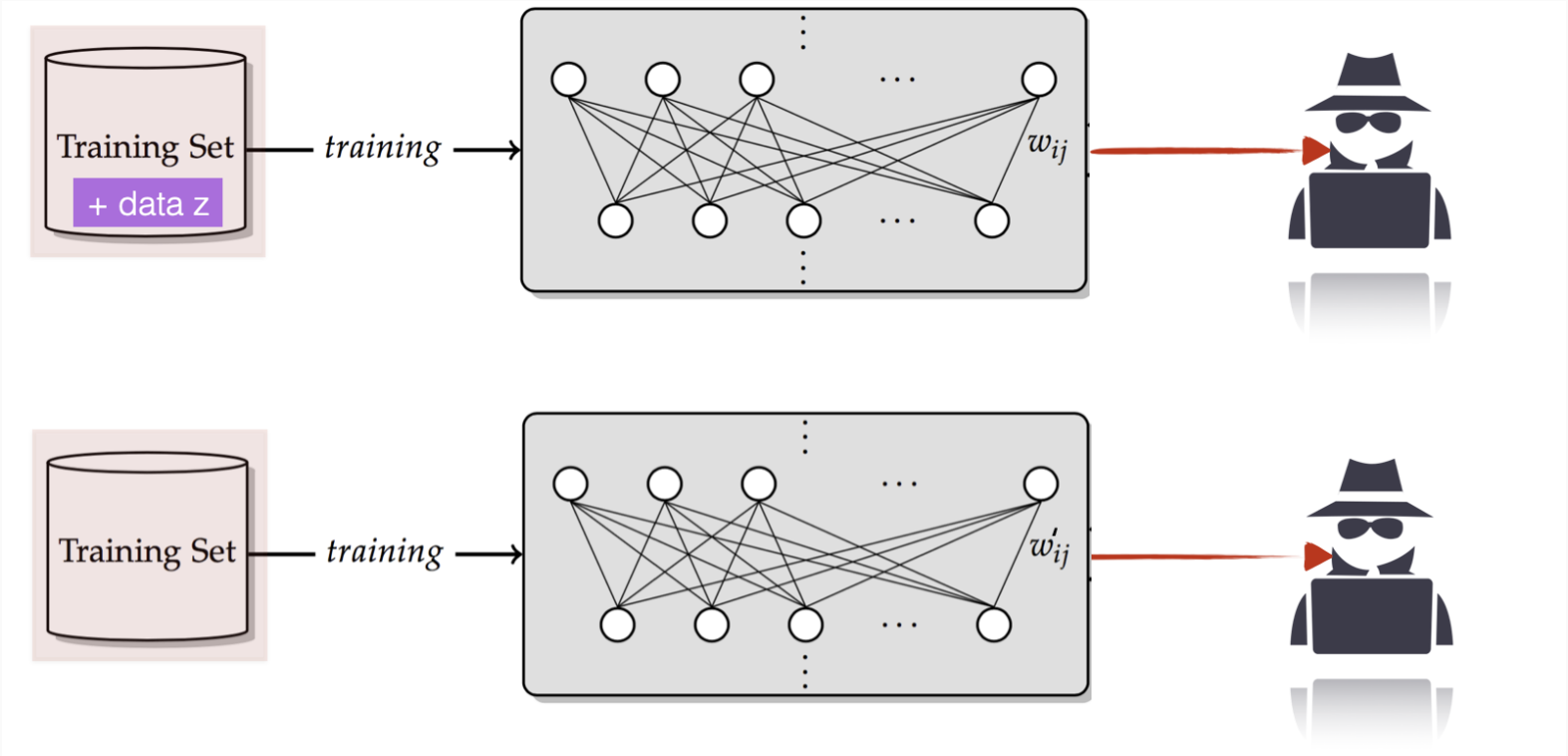
The focus is mostly on data collection, data sharing, access control, ...

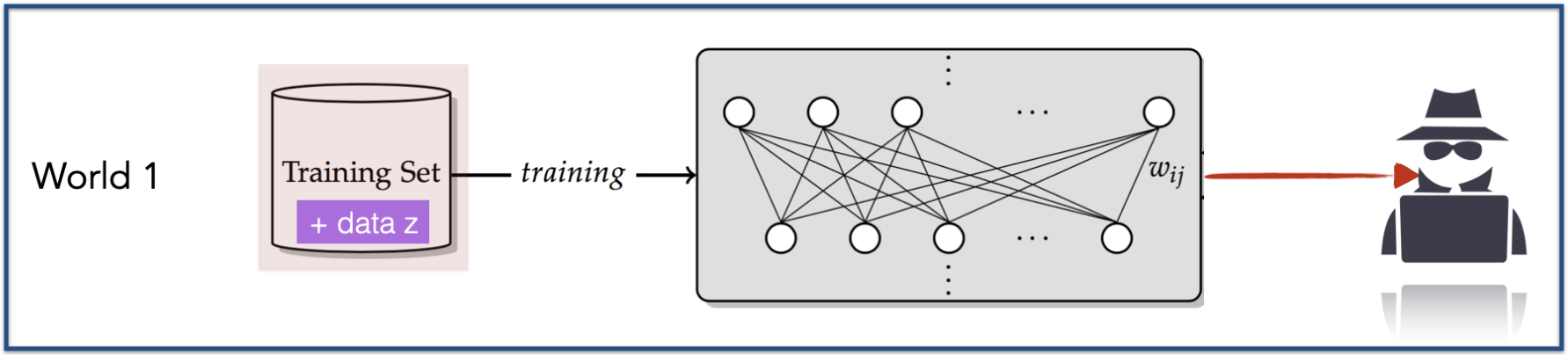
Indirect Privacy Risks

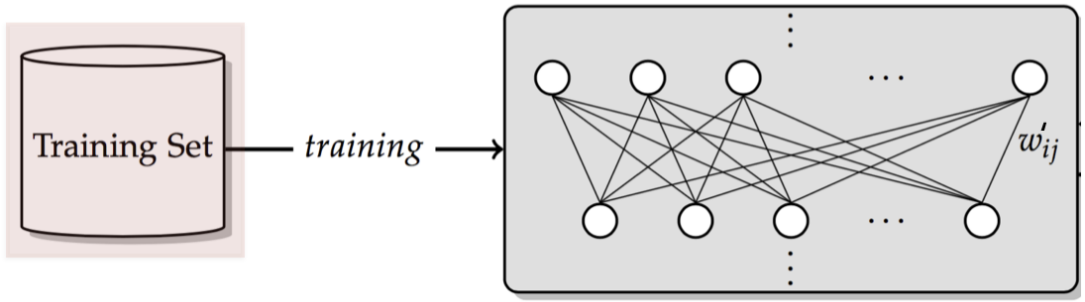


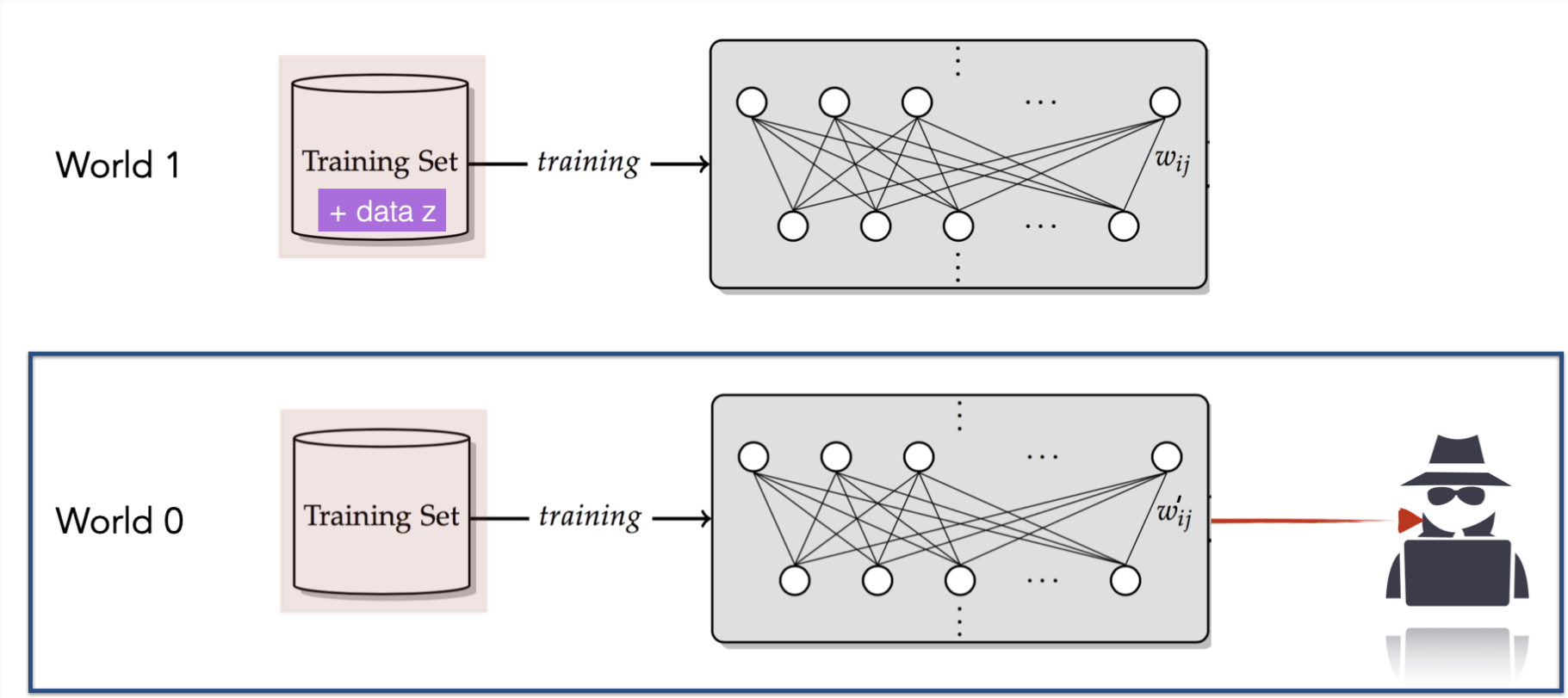
Models are **personal data** $\Rightarrow$ We need a standard method for quantitatively auditing data privacy in machine learning systems

Quantify privacy loss using Membership Inference Attacks


Training Set
+ data z
training
w_{ij}
Training Set
training
w'_{ij}

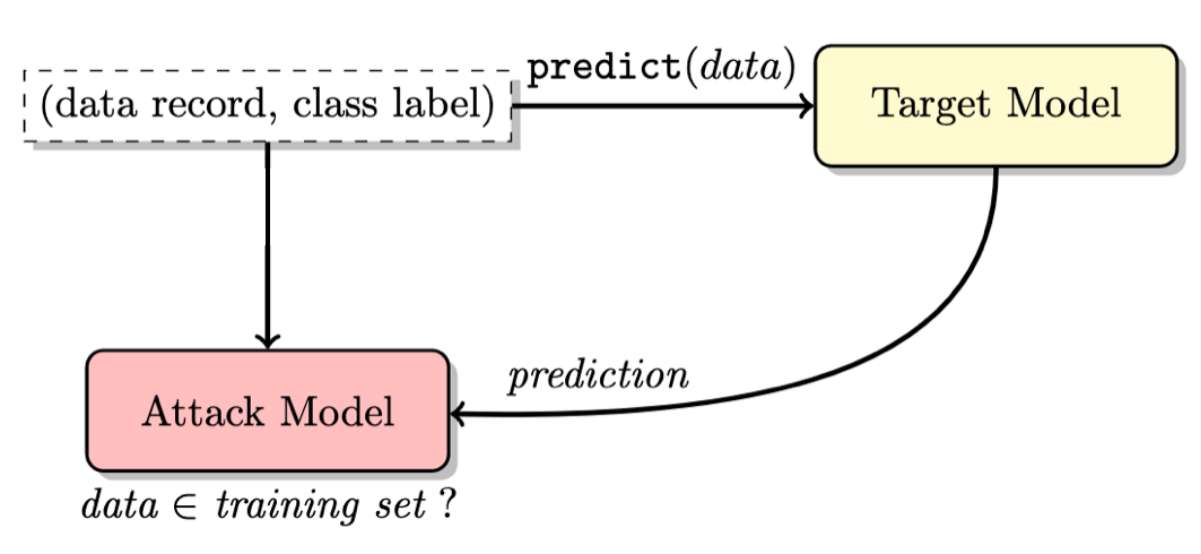

World 1
Training Set
+ data z
training
w_{ij}


World 0
Training Set
training
w'_{ij}


World 1
Training Set
+ data z
training
w_{ij}
World 0
Training Set
training
w'_{ij}

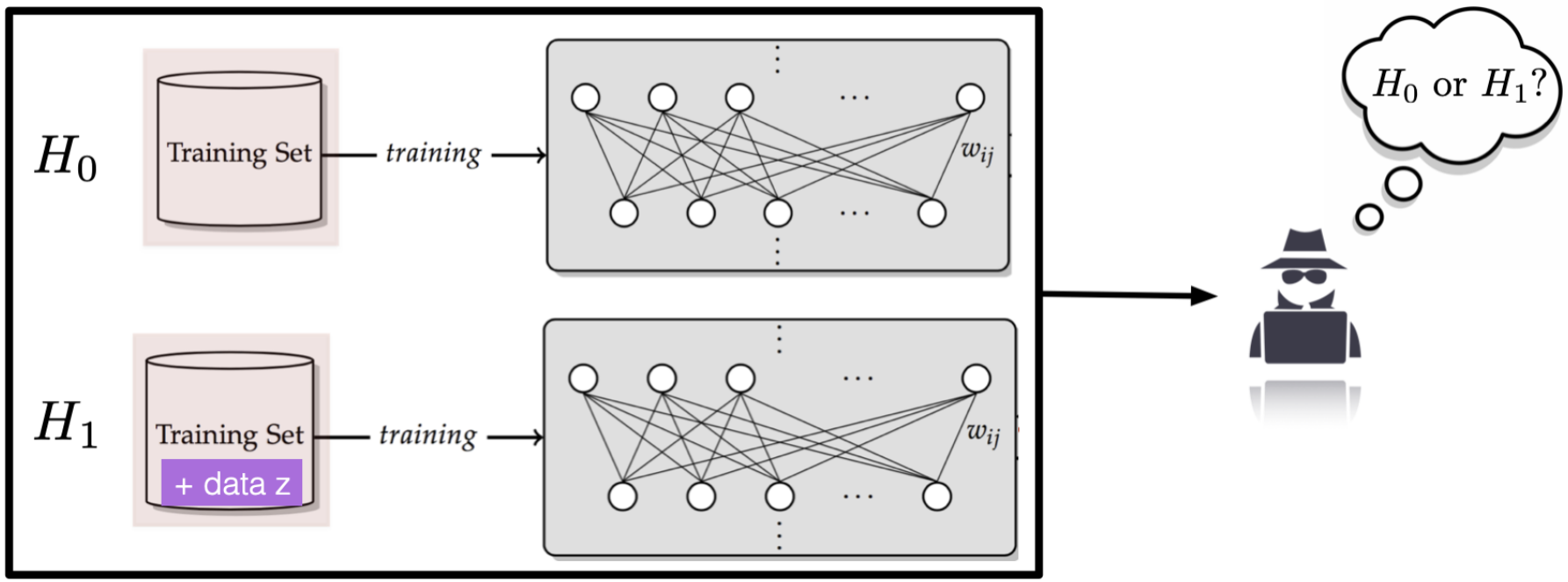
Membership Inference Attacks

- Given a model, can an adversary infer whether a particular data record is part of its training set?
- Success of attacker is a metric for privacy loss



Source: [Shokri et al., 2017]

Membership Inference Attack (MIA) Game

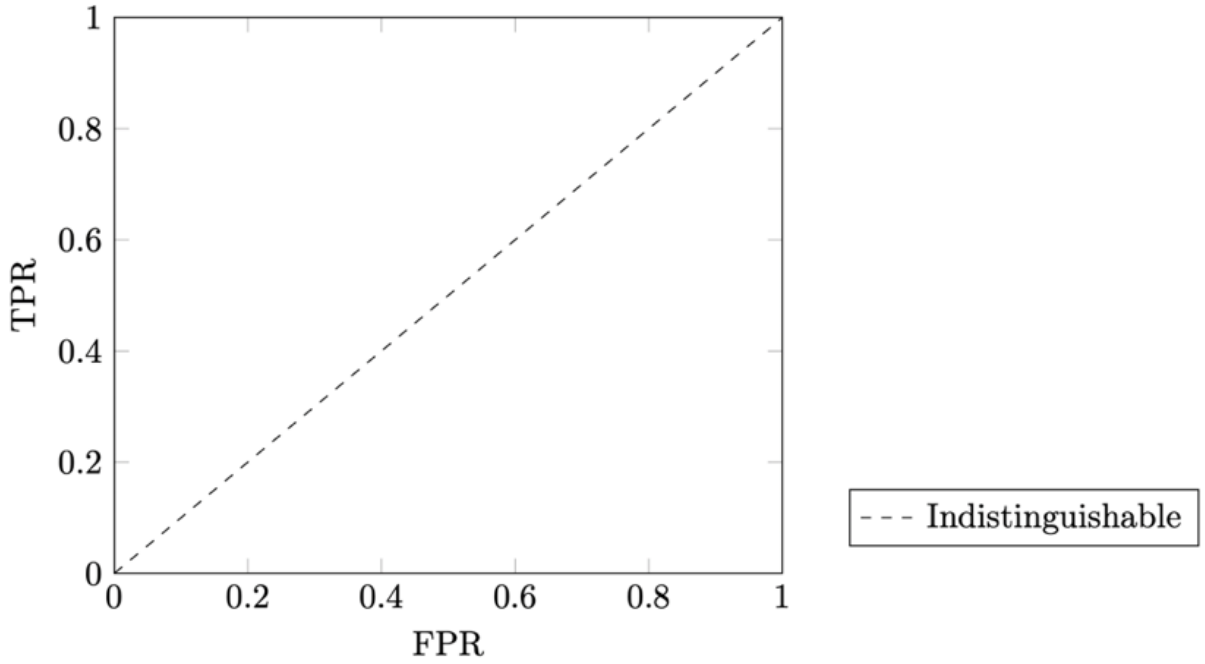


- Success of adversary indicates information leakage of models about their training data $\Rightarrow$ how to measure success in an informative way?

Quantifying Attack Success with Indistinguishability Metrics

Suppose that we run multiple random trials of the membership inference game (randomness is only over random coins of the training algorithm)

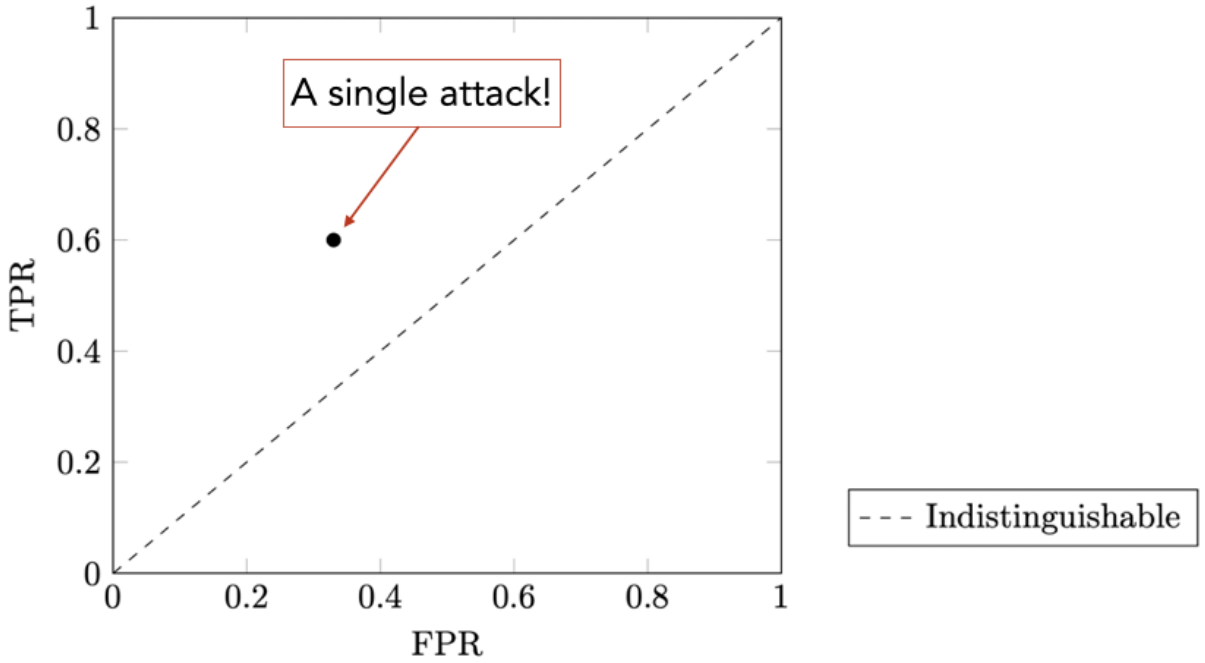
power: members are correctly predicted as member



error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

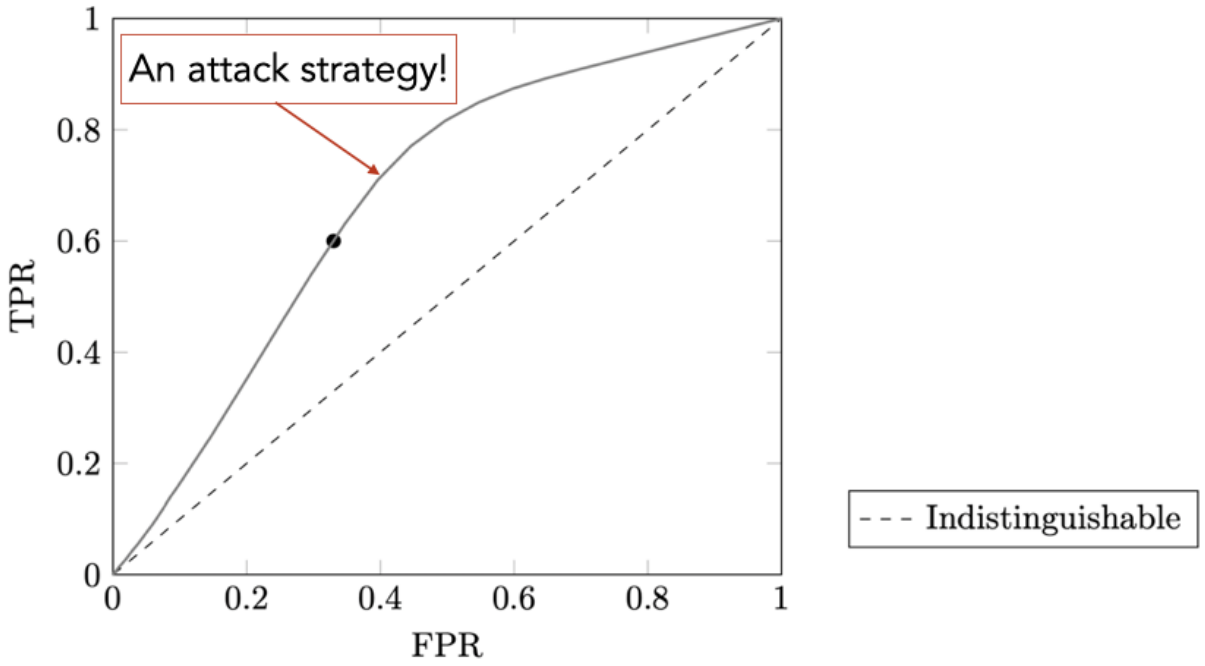
power: members are correctly predicted as member



error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

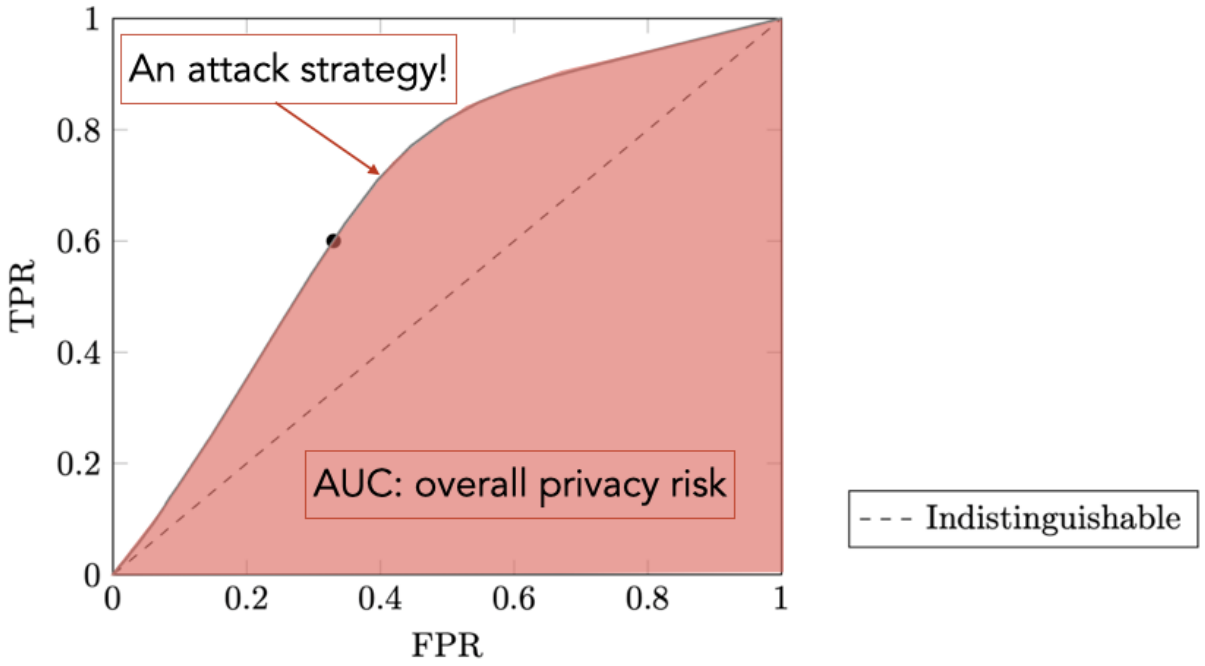
power: members are correctly predicted as member



error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

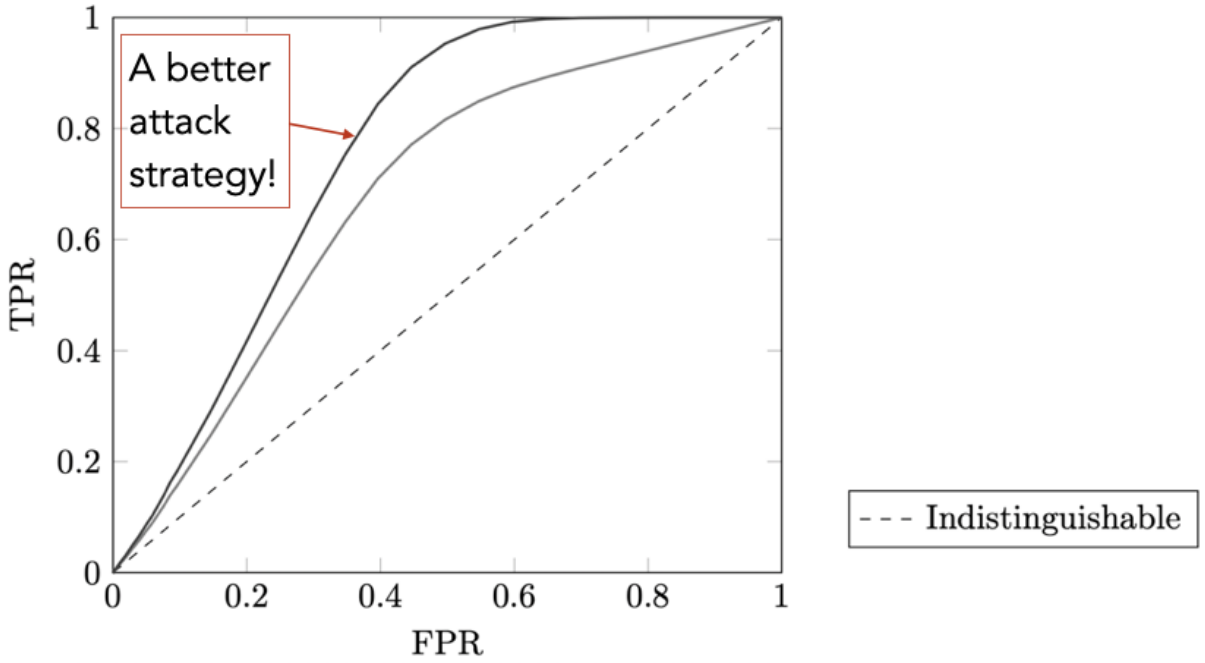
power: members are correctly predicted as member



error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

power: members are correctly predicted as member

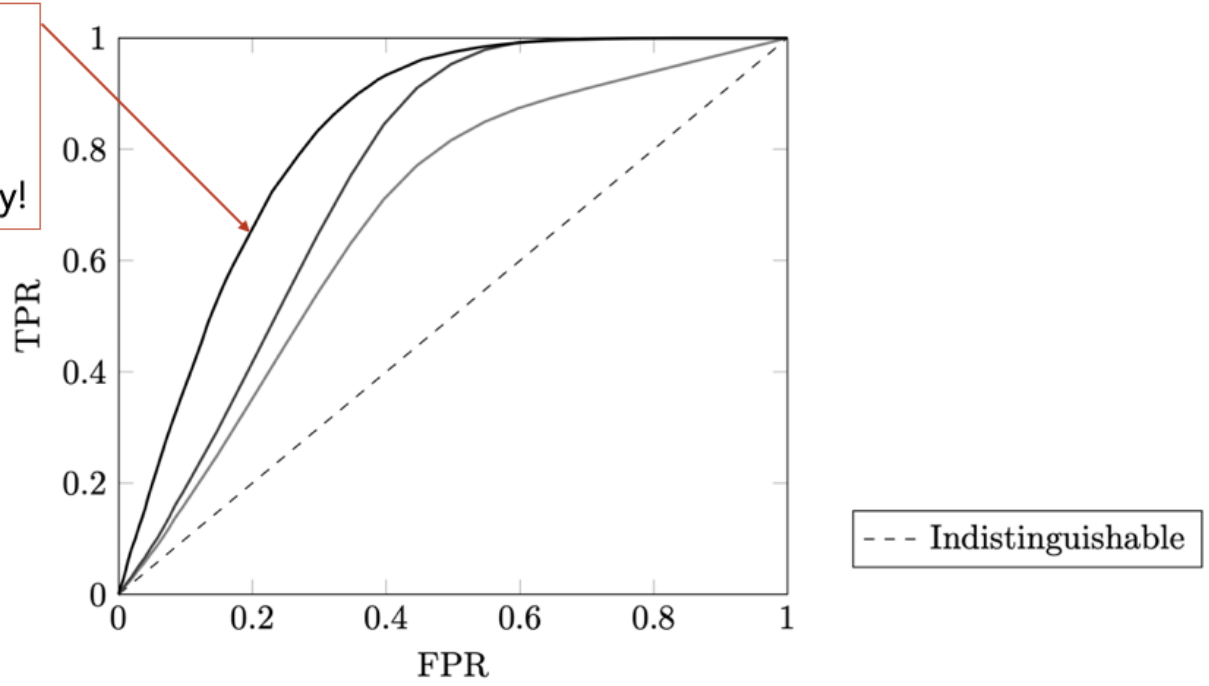


error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

Even a better attack strategy!

power: members are correctly predicted as member

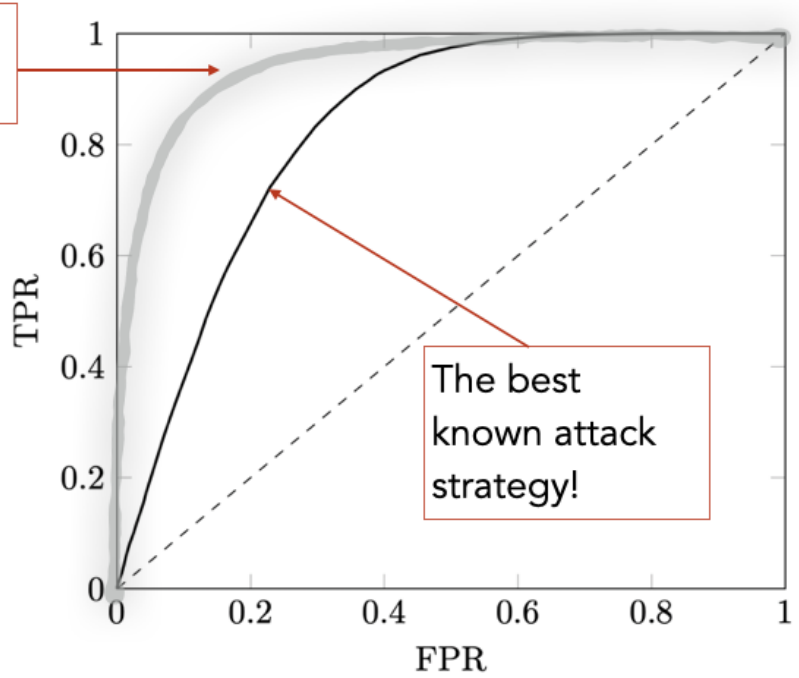


error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

An (unknown) **optimal** attack strategy

power: members are correctly predicted as member

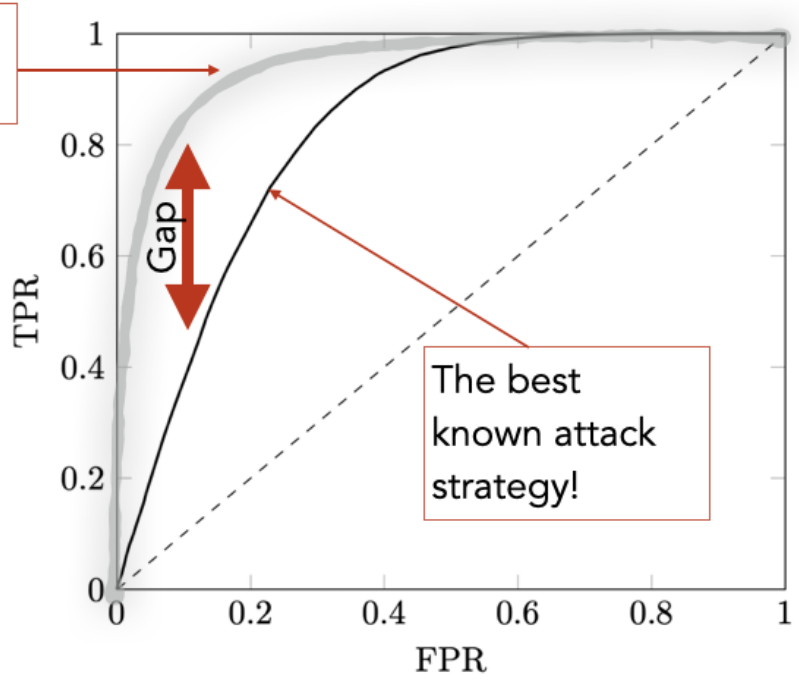


error: non-members are wrongly predicted as member

Quantifying Attack Success with Indistinguishability Metrics

An (unknown) **optimal** attack strategy

power: members are correctly predicted as member



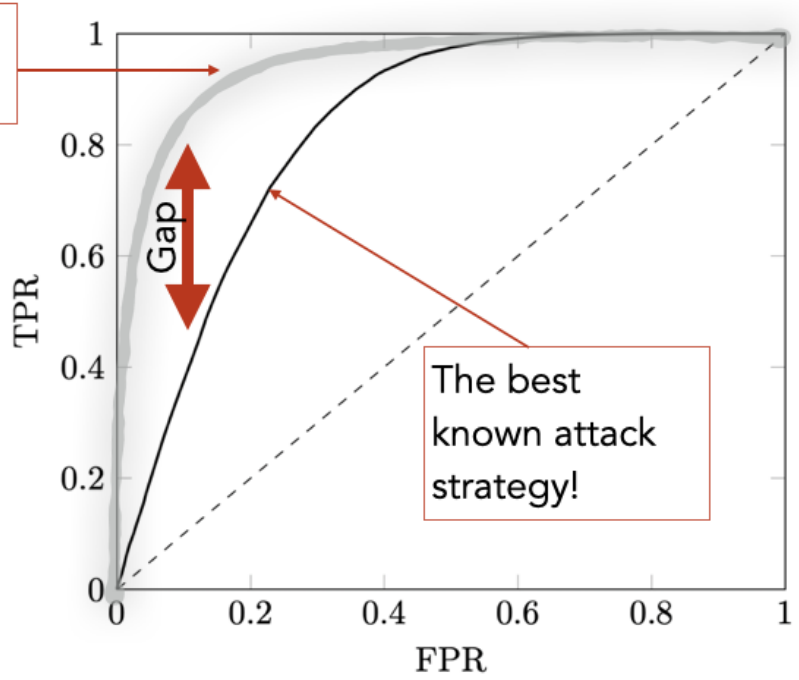
error: non-members are wrongly predicted as member

An attack strategy gives a lower-bound on the privacy risk of the target algorithm $\Rightarrow$ This is very useful to rule out vulnerable algorithms

Quantifying Attack Success with Indistinguishability Metrics

An (unknown) **optimal** attack strategy

power: members are correctly predicted as member



error: non-members are wrongly predicted as member

An attack strategy gives a lower-bound on the privacy risk of the target algorithm $\Rightarrow$ This is very useful to rule out vulnerable algorithms

Quantifying privacy risk with Attack Success

An attack strategy gives a lower-bound on the privacy risk of the target algorithm.

- This is very useful to rule out vulnerable algorithms,
- But, lack of a known powerful attack is not a guarantee for privacy!

We need to get close to the optimal attack strategy or to guarantee that the privacy risk under any attack never exceeds an upper bound

Quantifying privacy risk with Attack Success

An attack strategy gives a lower-bound on the privacy risk of the target algorithm.

- This is very useful to rule out vulnerable algorithms,
- But, lack of a known powerful attack is not a guarantee for privacy!

We need to get close to the optimal attack strategy or to guarantee that the privacy risk under any attack never exceeds an upper bound

How to design powerful inference attacks

How to design powerful inference attacks?

- The uncertainty of adversary depends on the data sampling process and the algorithm.
- For a powerful attack, we need to minimize the uncertainty of adversary to only the presence or absence of the target data

Source: [Murakonda et al., 2021, Ye et al., 2022]

A Simple Membership Inference Attack

Attack: If $\ell(\theta, x_z, y_z) \leq c_\alpha(\theta, x_z, y_z)$, predict Member

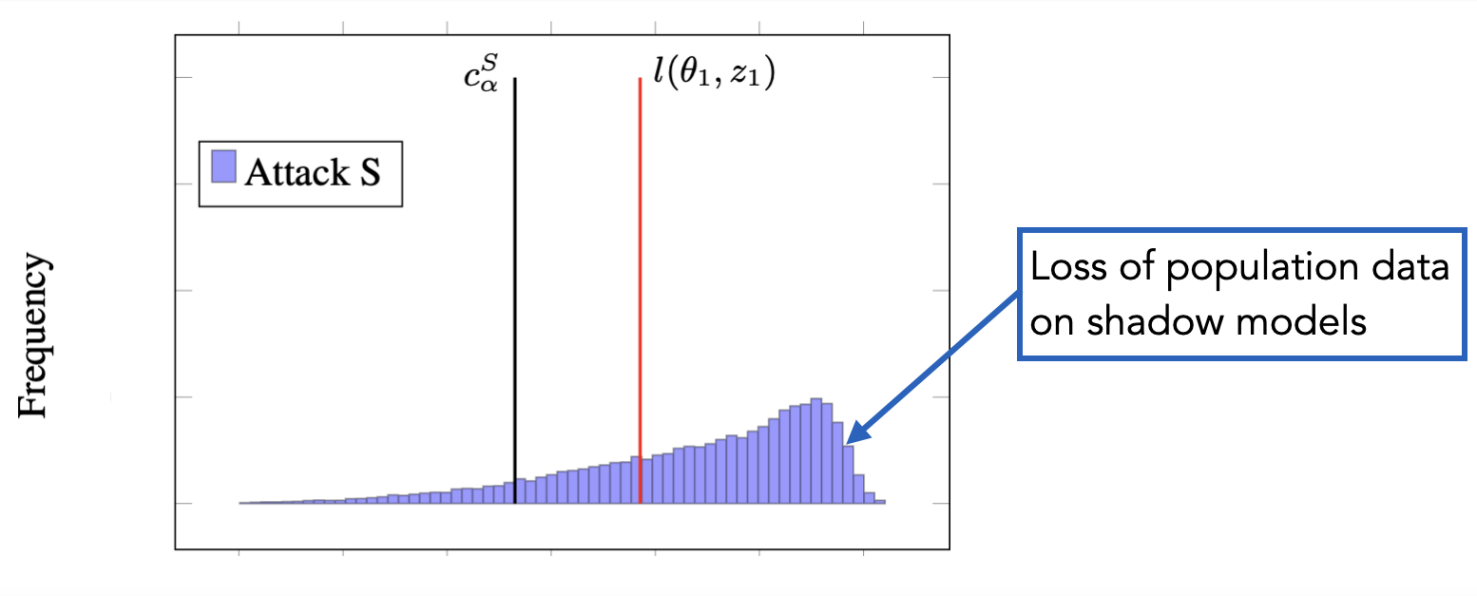
$\uparrow$

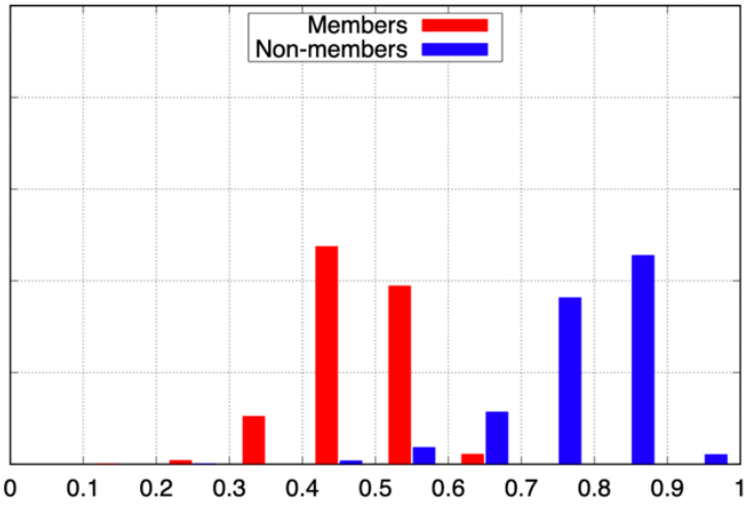
false positive rate

Membership Inference via Shadow Models

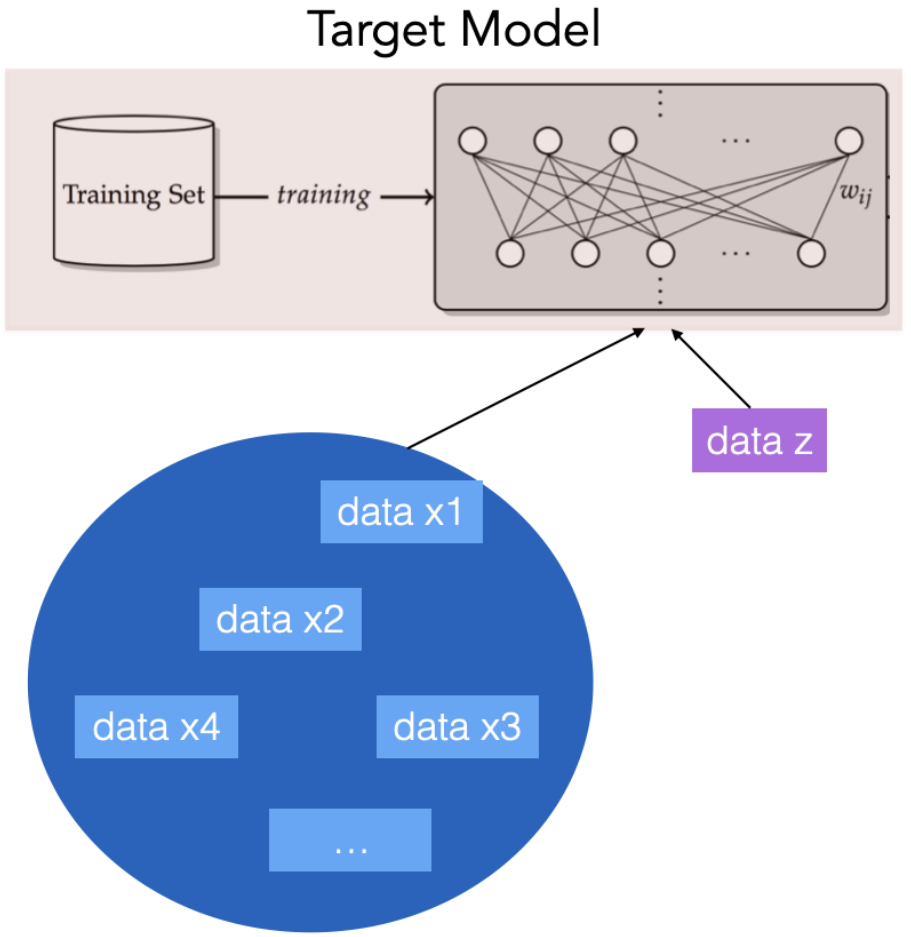
$$\text{If } \ell(\theta, x_z, y_z) \leq c_\alpha(y_z), \text{ predict Member}$$

- A large body of the literature is based on this technique [Shokri et al., 2017]
- Learn a threshold from the behavior of shadow models on their test data





Model's Error

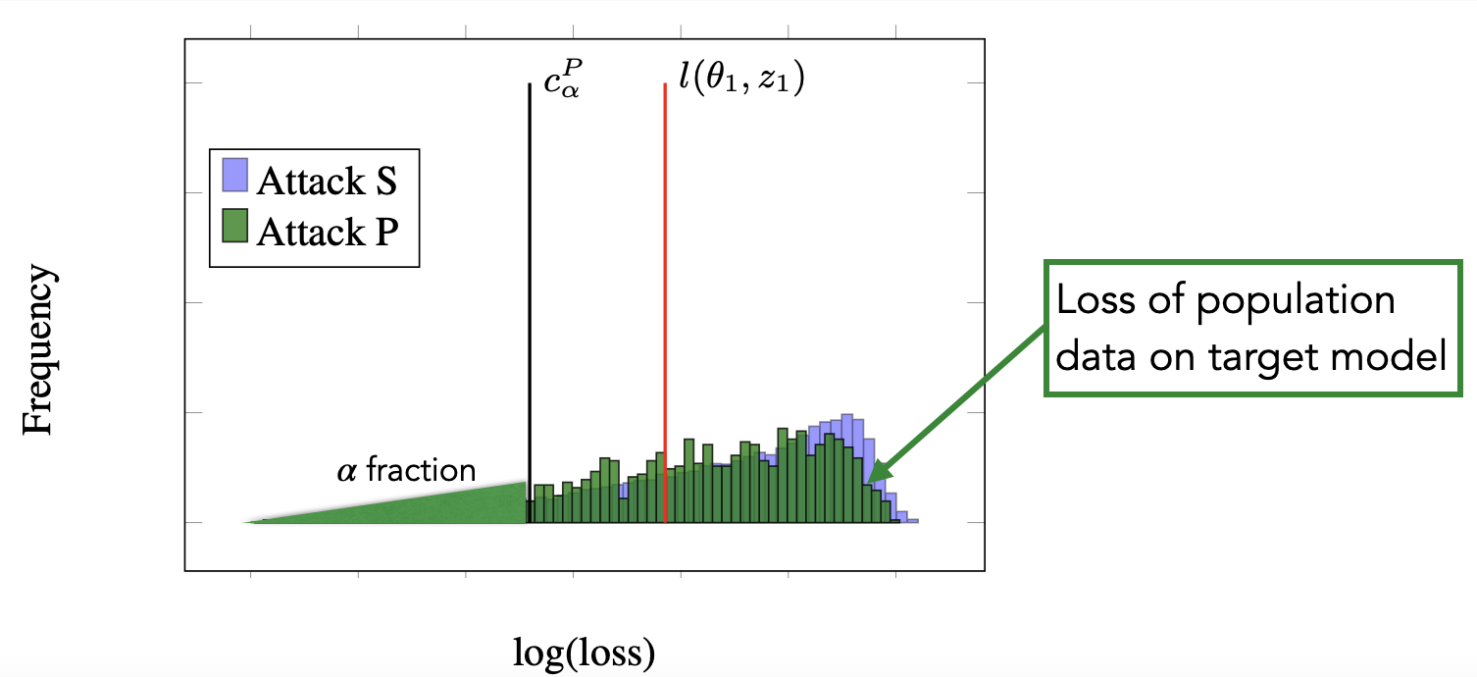


Source: [Shokri et al., 2017, Ye et al., 2022]

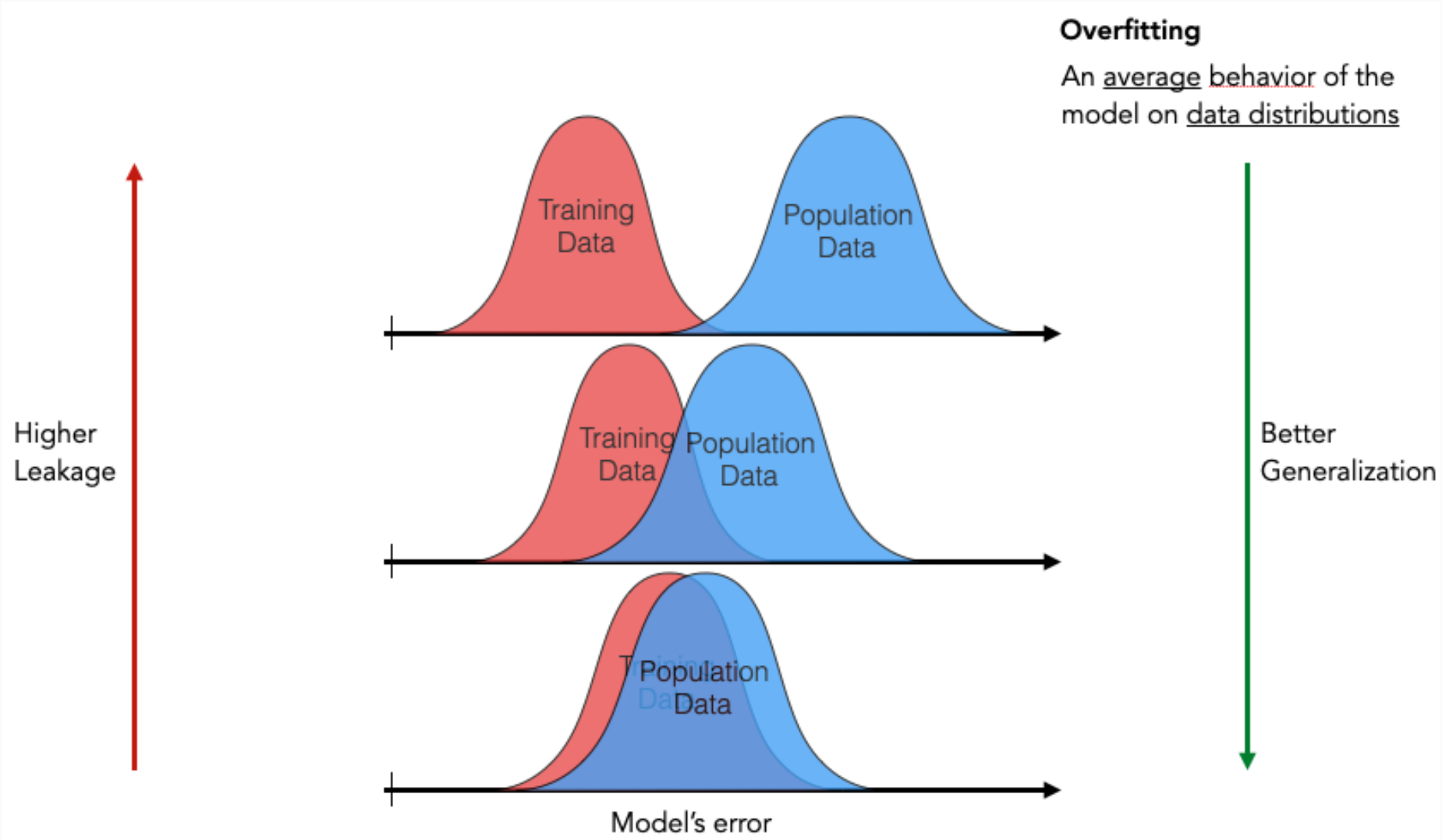
Membership Inference via Population Data

$$\text{If } \ell(\theta, x_z, y_z) \leq c_\alpha(\theta), \text{ predict Member}$$

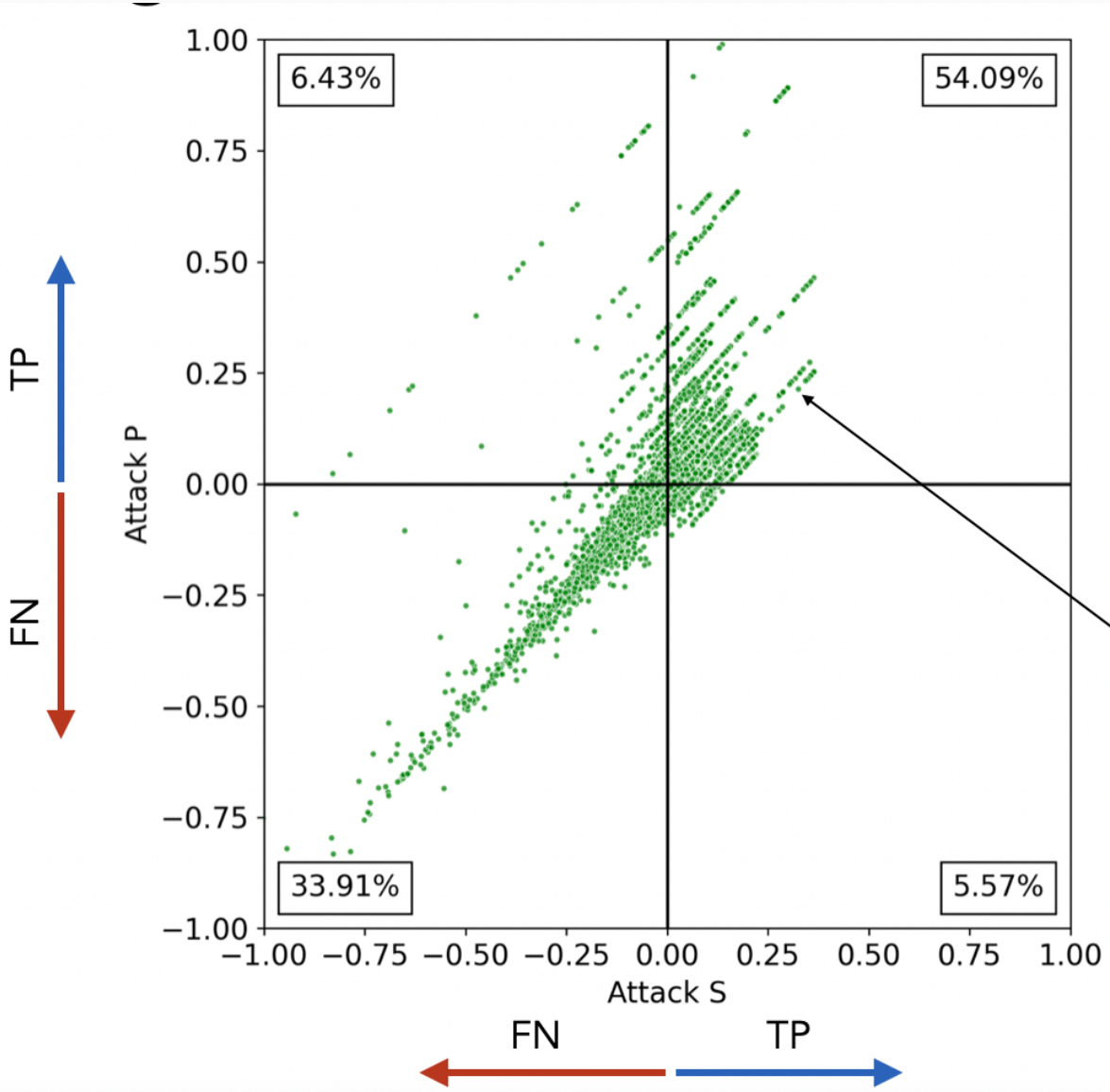
- Directly learn a threshold from the loss distribution of the target model on population data



Reason for Leakage?



Agreement between Attacks

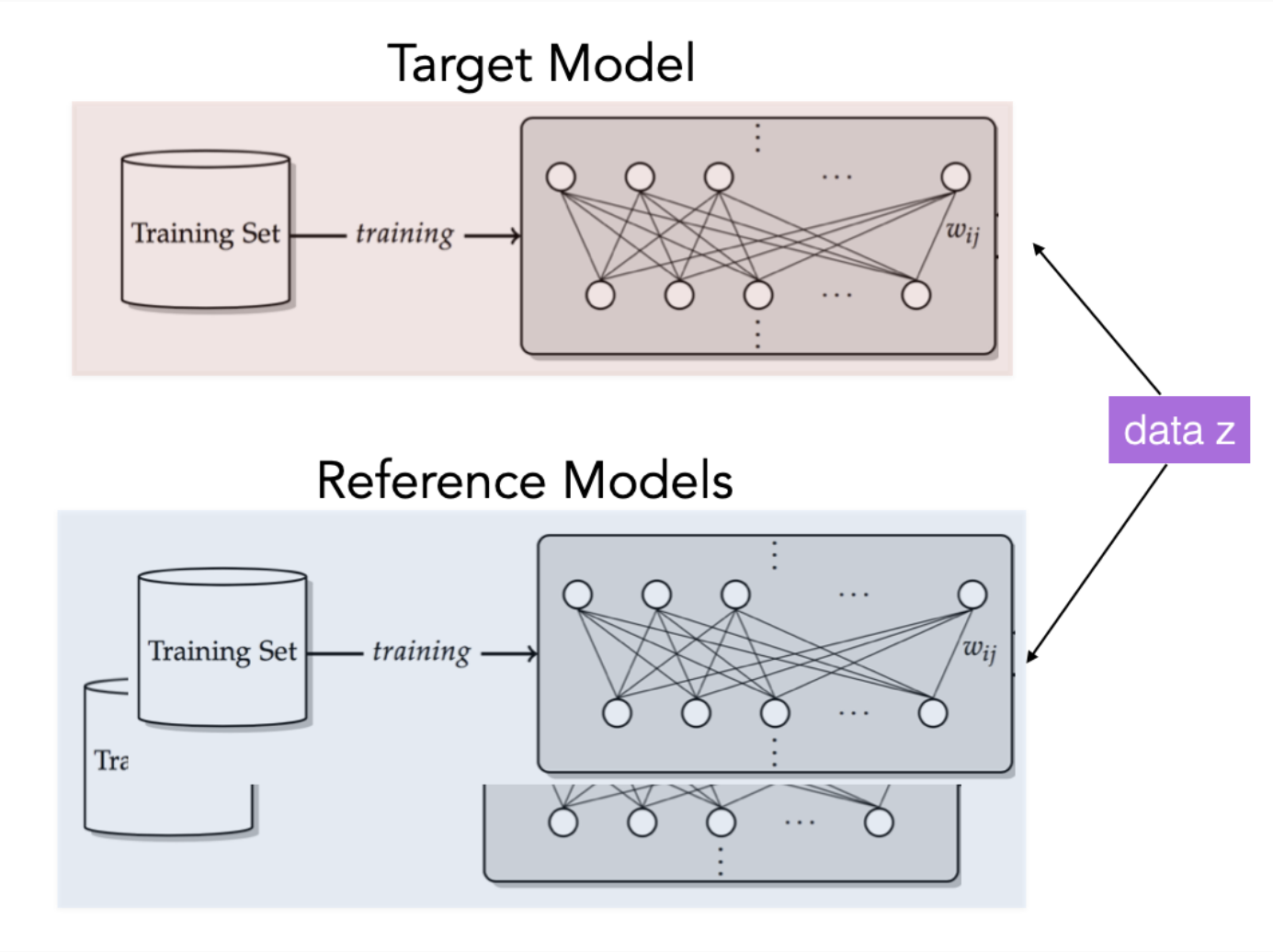


Distance of loss(z;theta) to attack thresholds computed on attacks P and S, for a given training data point z, and a fixed alpha (0.1)

How to perform a more accurate analysis?

- Leakage is not just an average property of the model, but rather the indistinguishability of the training algorithm's output when it is trained on any given data point versus when it is not

Reference Models

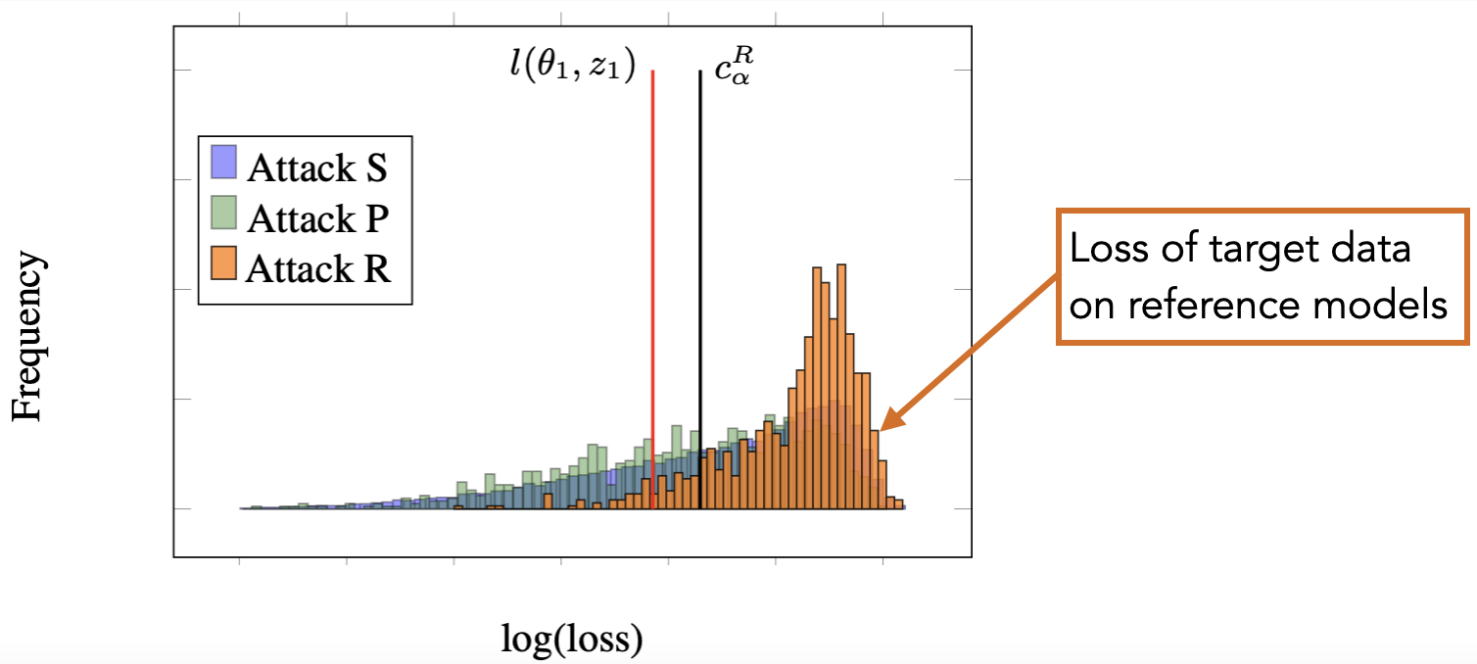


Source: [Ye et al., 2022]

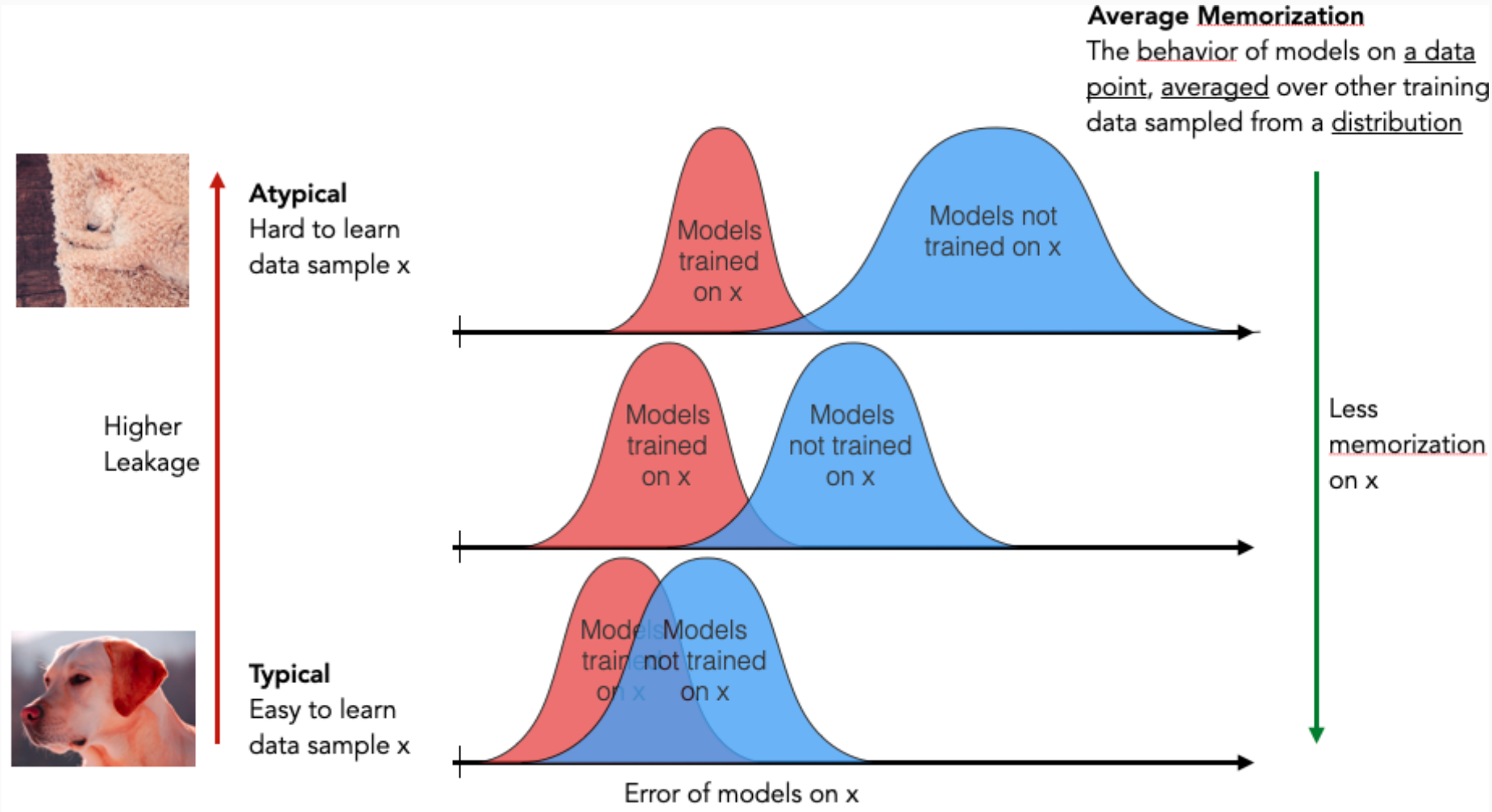
Membership Inference via Reference Models

$$\text{If } \ell(\theta, x_z, y_z) \leq c_\alpha(x_z, y_z), \text{ predict Member}$$

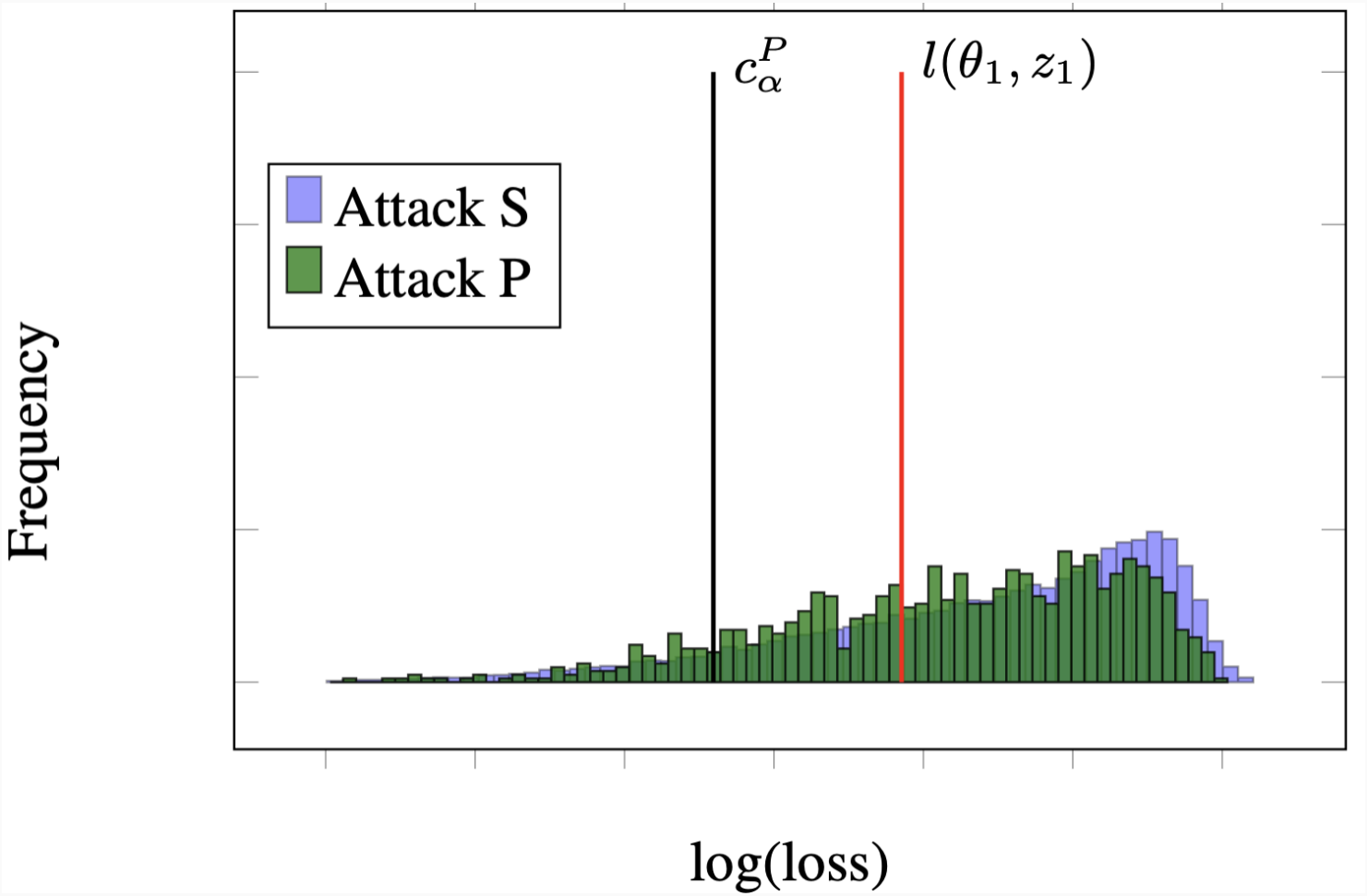
- Learn a threshold from the loss distribution of the target data on reference models



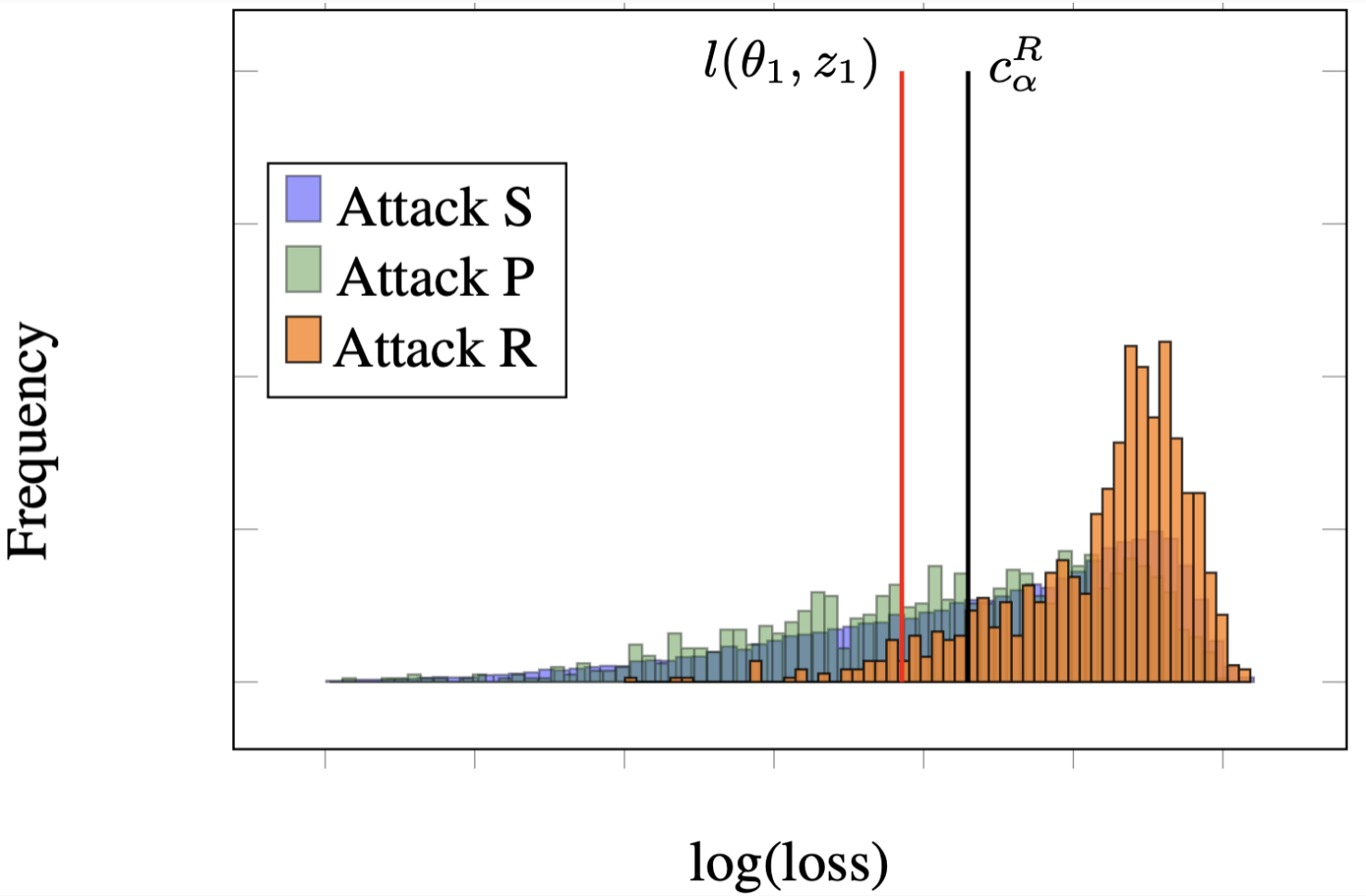
Reason for Leakage?



Comparing thresholds between Attacks



Comparing thresholds between Attacks



Design Low-Cost High-Power Membership Inference Attacks

Membership Inference Game

Let π denote the underlying distribution for the population data, and let T be the training algorithm. Let $x \sim \pi$ be the **target data point**.

- The challenger samples a dataset $D \sim \pi$ and a data point $z \sim \pi$.
- The challenger flips a fair coin b. The challenger trains a model θ on
  - $\{x\} \cup D$, if $b = 1$
  - $\{z\} \cup D$, if $b = 0$
- The challenger sends the **target model** θ to the adversary.
- The adversary, having access to the distribution over the population data π, outputs a membership bit $\hat{b} \leftarrow \mathrm{MIA}(x; \theta)$.

Likelihood Ratio Test i

We perform likelihood ratio tests to model the strategy of the adversary for this membership inference game.

$$\mathrm{LR}_\theta(x, z; D) = \frac{\Pr(\theta|x, D)}{\Pr(\theta|z, D)} \tag{1}$$

Given $\mathrm{LR}_\theta(x, z; D)$, the hypothesis test for our attack, which is a test for violation of data privacy, can be formulated as follows:

$$\mathrm{MIA}(x; \theta) = \Pr_{z\sim\pi, D\sim\pi}\left(\mathrm{LR}_\theta(x, z; D) \geq \gamma\right) \geq \beta \tag{2}$$

Likelihood Ratio Test i

We perform likelihood ratio tests to model the strategy of the adversary for this membership inference game.

$$\mathrm{LR}_\theta(x, z; D) = \frac{\Pr(\theta|x, D)}{\Pr(\theta|z, D)} \tag{1}$$

Given $\mathrm{LR}_\theta(x, z; D)$, the hypothesis test for our attack, which is a test for violation of data privacy, can be formulated as follows:

$$\mathrm{MIA}(x; \theta) = \Pr_{z\sim\pi, D\sim\pi}\left(\mathrm{LR}_\theta(x, z; D) \geq \gamma\right) \geq \beta \tag{2}$$

Likelihood Ratio Test ii

$$
\begin{aligned}
&\mathrm{LR}_\theta(x, z; D) \\
&= \frac{\Pr(\theta|x, D)}{\Pr(\theta|z, D)} \\
&= \frac{\left(\frac{\Pr(x|\theta)\Pr(D|\theta)\Pr(\theta)}{\int_{\theta'}\Pr(x|\theta')\Pr(D|\theta')\Pr(\theta')\,\mathrm{d}\theta'}\right)}{\left(\frac{\Pr(z|\theta)\Pr(D|\theta)\Pr(\theta)}{\int_{\theta'}\Pr(z|\theta')\Pr(D|\theta')\Pr(\theta')\,\mathrm{d}\theta'}\right)}
\end{aligned}
\tag{3}
$$

We can empirically calculate the normalizing denominators by sampling θ' models (which we will refer to as *reference models*), each trained on data randomly drawn from the population distribution π. Through this empirical process, $\Pr(D|\theta')$ achieves values that are almost the same across the sampled θ' models, allowing it to be approximated by a constant (i.e., the prediction distribution on IID samples from the population distribution remains consistent among models trained on IID samples from that same distribution).

Likelihood Ratio Test iii

The LR can be further simplified and approximated as:

$$\mathrm{LR}_\theta(x, z) = \left(\frac{\Pr(x|\theta)}{\Pr(x)}\right) \cdot \left(\frac{\Pr(z|\theta)}{\Pr(z)}\right)^{-1} \tag{4}$$

where, $\Pr(x) = \frac{1}{n}\sum_{\theta'} \Pr(x|\theta')$ (and similarly for $\Pr(z)$) which is the empirical average of $\Pr(x|\theta')$ on reference models θ' trained on data sampled from π. The quantity $\Pr(x)$ reflects the average probability of a data point in the distribution of models, regardless of whether or not it is part of their training data. The quantity $\Pr(x|\theta)$ reflects the probability of a data point according to the target model θ. This is the probability that training algorithms aim to maximize over the training data in a maximum likelihood estimation.

Likelihood Ratio Test iv

Given the $\mathrm{LR}_\theta(x, z)$ computation in (4), we can estimate our membership inference test as:

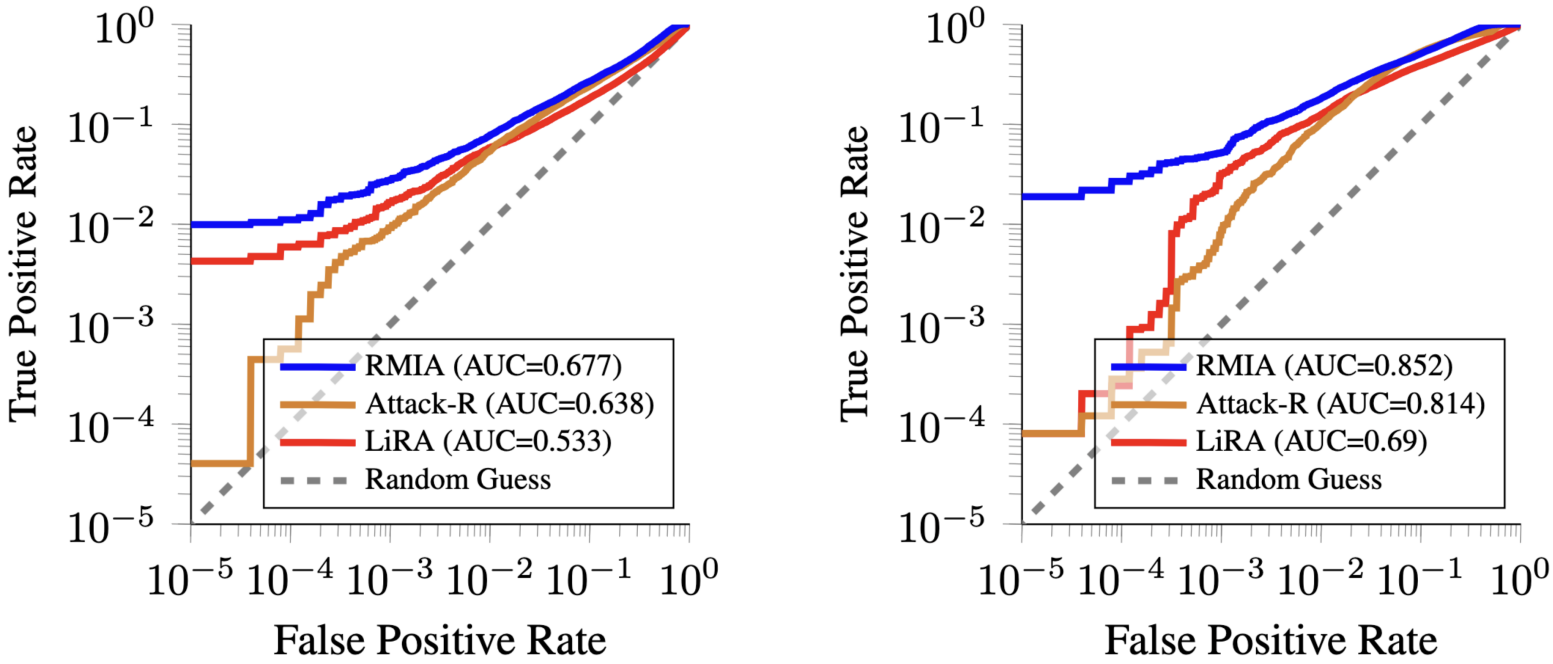
$$\mathrm{MIA}(x; \theta) = \Pr_{z \sim \pi} \left(\mathrm{LR}_\theta(x, z) \geq \gamma \right) \geq \beta \tag{5}$$

Membership Inference Attack

- Input: model θ, data point x, and test parameters γ, β.
- Sample many $z \sim \pi$, and compute the fraction δ of z samples that pass the relative membership inference likelihood ratio test $\mathrm{LR}_\theta(x, z) \geq \gamma$. [See (4)]
- Return member if $\delta \geq \beta$, otherwise, return non-member. [See (5)]

This test offers an interpretable membership inference attack, wherein the threshold β is used to gauge how distinguishable a data point x is from the rest of the population data in terms of their influence on θ.

Attack Results



(a) CIFAR-10 (b) CIFAR-100

Figure 1: The performance comparison between our attack (RMIA) and the prior works (including LiRA (Carlini et al., 2022) and also Attack-R (Ye et al., 2022), under computation constraints, with the restriction of using only 1 reference model, for attacking one single model.

Quantifying privacy risk with Attack Success

An attack strategy gives a lower-bound on the privacy risk of the target algorithm.

- This is very useful to rule out vulnerable algorithms,
- But, lack of a known powerful attack is not a guarantee for privacy!

How to guarantee that the privacy risk under any attack never exceeds an upper bound?

Quantifying privacy risk with Attack Success

An attack strategy gives a lower-bound on the privacy risk of the target algorithm.

- This is very useful to rule out vulnerable algorithms,
- But, lack of a known powerful attack is not a guarantee for privacy!

How to guarantee that the privacy risk under any attack never exceeds an upper bound?

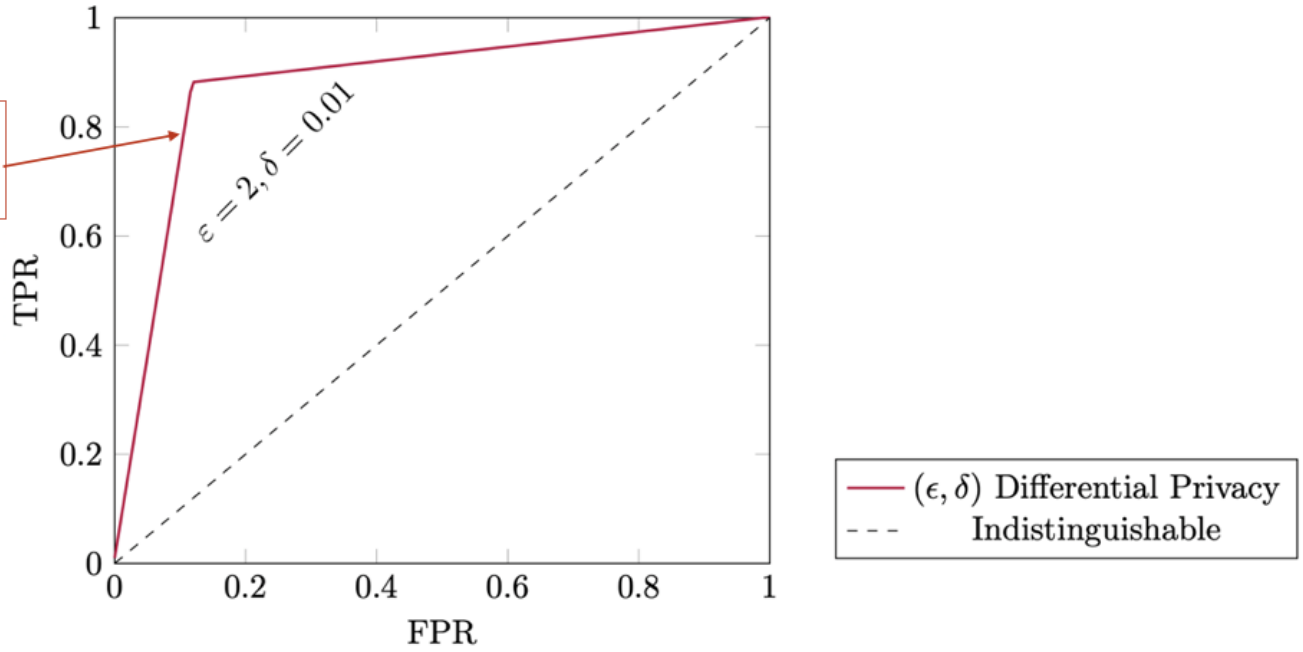
How to guarantee privacy (differentially private algorithms)

How can we guarantee privacy?

Prove an **upper-bound** for the privacy risk of a randomized algorithm

An Upper bound on privacy loss

$$e^{\epsilon} \cdot (1 - TPR) + FPR \geq 1 - \delta$$
$$1 - TPR + e^{\epsilon} \cdot FPR \geq 1 - \delta$$

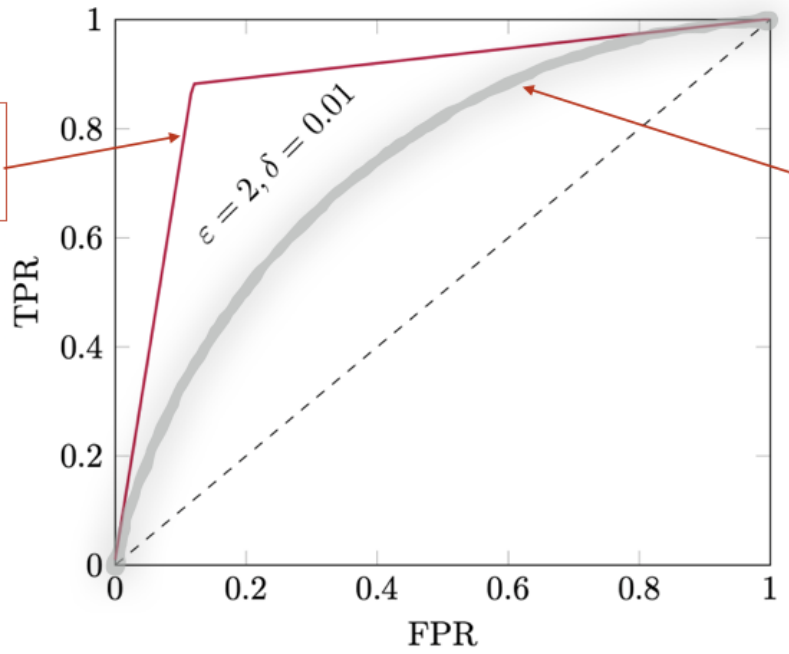


How can we guarantee privacy?

Prove an **upper-bound** for the privacy risk of a randomized algorithm

An Upper bound on privacy loss

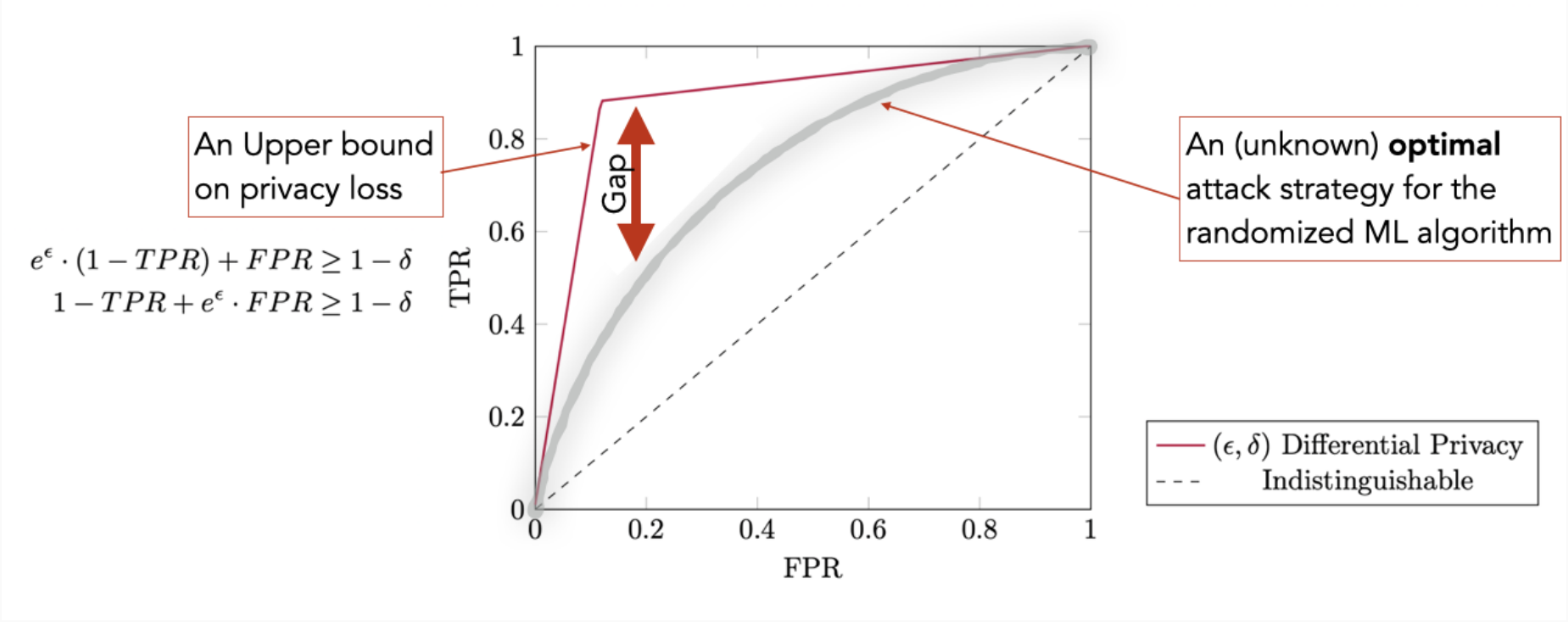
$$e^{\epsilon} \cdot (1 - TPR) + FPR \geq 1 - \delta$$
$$1 - TPR + e^{\epsilon} \cdot FPR \geq 1 - \delta$$



An (unknown) **optimal** attack strategy for the randomized ML algorithm

How can we guarantee privacy?

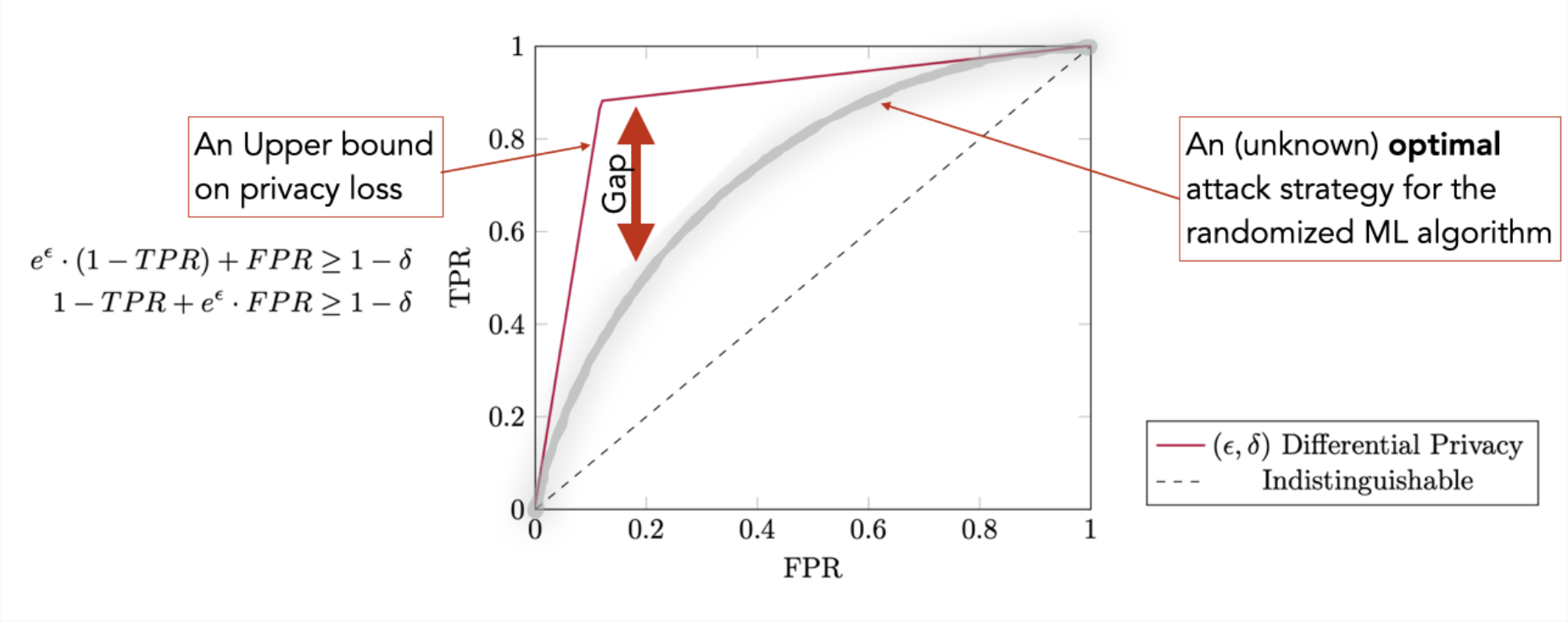
Prove an **upper-bound** for the privacy risk of a randomized algorithm



- A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

How can we guarantee privacy?

Prove an **upper-bound** for the privacy risk of a randomized algorithm



- A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} \cdot \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

Setting

- X (sensitive data) $\longrightarrow$ mechanism $\longrightarrow Y$ (observables)
- Attacker
  - observes Y
  - knows the mechanism and has some background knowledge about X,
  - wants to infer secret information about X

$$\Pr[X|Y] = \frac{\overbrace{\Pr[Y|X]}^{\text{mechanism}} . \overbrace{\Pr[X]}^{\text{knowledge}}}{\Pr[Y]}$$

- Fact: we cannot control adversary's background knowledge
- Objective: design a mechanism that limits information leakage about data, basically by controlling $\Pr[Y|X]$

A Strict Definition for Privacy

- Perfect indistinguishability: For all inputs, the output probability is the same.

$$\forall x, x', y: \qquad \Pr[Y = y|X = x] = \Pr[Y = y|X = x']$$

- The mechanism does not leak any information about X
  - However, achieving it is very hard $\Rightarrow$ may produce pseudorandom outputs (not useful information about the data)

A Strict Definition for Privacy

- Perfect indistinguishability: For all inputs, the output probability is the same.

$$\forall x, x', y: \qquad \Pr[Y = y | X = x] = \Pr[Y = y | X = x']$$

- The mechanism does not leak any information about X
  - However, achieving it is very hard $\Rightarrow$ may produce pseudorandom outputs (not useful information about the data)

A Strict Definition for Privacy

- Perfect indistinguishability: For all inputs, the output probability is the same.

$$\forall x, x', y: \qquad \Pr[Y = y|X = x] = \Pr[Y = y|X = x']$$

- The mechanism does not leak any information about X
  - However, achieving it is very hard $\Rightarrow$ may produce pseudorandom outputs (not useful information about the data)

A better definition for Privacy

- Some indistinguishability: For all similar inputs, the difference in output probabilities is bounded.

$$\forall y, \forall \text{ similar } x, x' : \qquad \frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']} \leq \text{ constant}$$

- It means by observing any y, adversary is NOT able to distinguish between inputs x and x' beyond a bounded certainty
- What does similar mean? Consider, for example, location positions that are within a range, or datasets that differ in one record, etc.

A better definition for Privacy

- Some indistinguishability: For all similar inputs, the difference in output probabilities is bounded.

$$\forall y, \forall \text{ similar } x, x' : \qquad \frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']} \leq \text{ constant}$$

- It means by observing any y, adversary is NOT able to distinguish between inputs x and x' beyond a bounded certainty
- What does similar mean? Consider, for example, location positions that are within a range, or datasets that differ in one record, etc.

A better definition for Privacy

- Some indistinguishability: For all similar inputs, the difference in output probabilities is bounded.

$$\forall y, \forall \text{ similar } x, x' : \qquad \frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']} \leq \text{ constant}$$

- It means by observing any y, adversary is NOT able to distinguish between inputs x and x' beyond a bounded certainty
- What does similar mean? Consider, for example, location positions that are within a range, or datasets that differ in one record, etc.

Randomness

- Where is the source of randomness in $\Pr[Y|X]$?
- We design mechanisms that randomize data or the computations on data: randomized (input/output) perturbations

Randomness

- Where is the source of randomness in $\Pr[Y|X]$?
- We design mechanisms that randomize data or the computations on data: randomized (input/output) perturbations

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \not{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y|X = x]}{\Pr[Y = y|X = x']}\right) \leq \epsilon$$

(a) Large ϵ

(b) Small ε

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y|X = x]}{\Pr[Y = y|X = x']}\right) \leq \epsilon$$

(a) Large ϵ

(b) Small ε

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}\right) \leq \epsilon$$

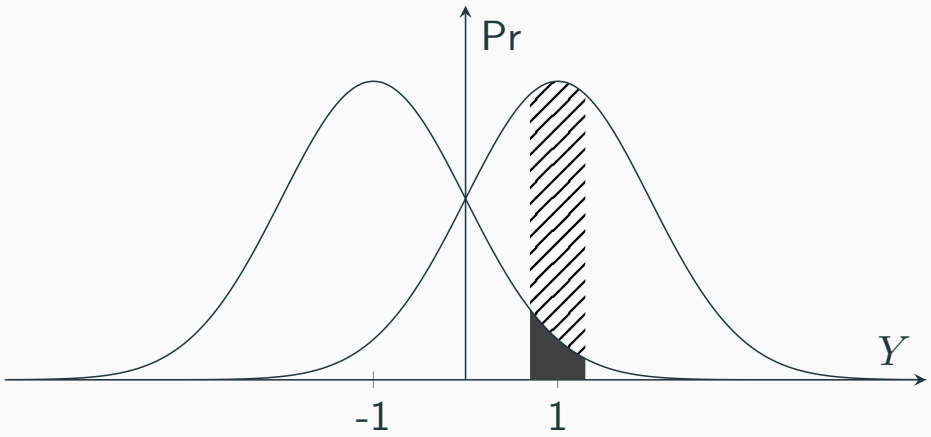
(a) Large ϵ

(b) Small ε

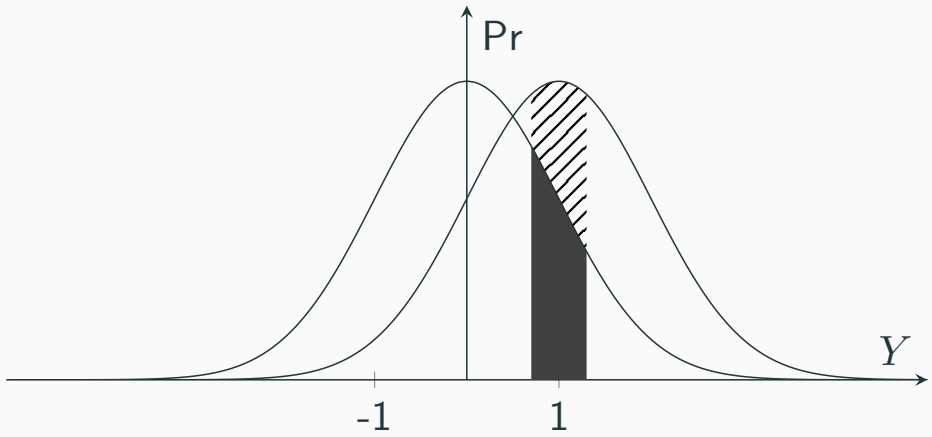
Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}\right) \leq \epsilon$$



(a) Large ϵ



(b) Small ε

An Approximate Notion of Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: (ϵ, δ)-DP

$$\forall x, x' : \qquad \Pr\left[\overbrace{\ln\left(\frac{\Pr[Y = y|X = x]}{\Pr[Y = y|X = x']}\right) > \epsilon}^{\text{violating } \epsilon\text{-DP}}\right] < \delta$$

  where the randomness of probability is over output y drawn from the output distribution $Pr[Y|X = x]$

- The chance that we have unbounded privacy loss is very small (δ)

An Approximate Notion of Differential Privacy

$$\Pr[Y = y|X = x] \leq e^{\epsilon} \Pr[Y = y|X = x'] + \delta$$

The promise of differential privacy

Theorem
Let $D, D \cup z$ be an arbitrary (worst-case) pair of neighboring datasets. If the algorithm $\mathcal{T}$ is (ε, δ)-differentially private, then the TPR and FPR of any attack algorithm $\mathcal{A}$, over random trials of the membership inference game, satisfy the following equation.

$$FPR + e^{\varepsilon} \cdot (1 - TPR) \geq 1 - \delta \tag{6}$$

$$e^{\varepsilon} \cdot FPR + (1 - TPR) \geq 1 - \delta \tag{7}$$

The proof is a direct application of the definition (left as assignment next week).

Source: [Wasserman and Zhou, 2010, Kairouz et al., 2015]

Summary: differential privacy guarantee

A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

How to compute the differential privacy bound of randomized algorithm?

- If the bound is loose, we are over-estimating the risk, thus we unnecessarily over-randomize the algorithm, ...
- This could result in a high utility drop (e.g., prediction error) in the algorithm.

Summary: differential privacy guarantee

A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

How to compute the differential privacy bound of randomized algorithm?

- If the bound is loose, we are over-estimating the risk, thus we unnecessarily over-randomize the algorithm, ...
- This could result in a high utility drop (e.g., prediction error) in the algorithm.

Summary: differential privacy guarantee

A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

How to compute the differential privacy bound of randomized algorithm?

- If the bound is loose, we are over-estimating the risk, thus we unnecessarily over-randomize the algorithm, ...
- This could result in a high utility drop (e.g., prediction error) in the algorithm.

Summary: differential privacy guarantee

A differential privacy guarantee is an **upper-bound** on the privacy risk of a randomized algorithm.

How to compute the differential privacy bound of randomized algorithm?

- If the bound is loose, we are over-estimating the risk, thus we unnecessarily over-randomize the algorithm, ...
- This could result in a high utility drop (e.g., prediction error) in the algorithm.

Next lecture: differentially private machine learning

- How to design differentially private learning algorithm?
- How to compute (tight) differential privacy upper bound?
- How to improve the trade-off between privacy and accuracy for learning algorithm?

References i


Kairouz, P., Oh, S., and Viswanath, P. (2015).
The composition theorem for differential privacy.
In International conference on machine learning, pages 1376–1385. PMLR.


Murakonda, S. K., Shokri, R., and Theodorakopoulos, G. (2021).
Quantifying the privacy risks of learning high-dimensional graphical models.
In International Conference on Artificial Intelligence and Statistics, pages 2287–2295. PMLR.

References ii

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017).
Membership inference attacks against machine learning models.
In 2017 IEEE symposium on security and privacy (SP), pages 3–18. IEEE.

Wasserman, L. and Zhou, S. (2010).
A statistical framework for differential privacy.
Journal of the American Statistical Association, 105(489):375–389.

References iii


Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. (2022).
Enhanced membership inference attacks against machine learning models.
In ACM Conference on Computer and Communications Security (CCS).