Lecture 1: Privacy I

Franziska Boenisch and Adam Dziedzic

Course on Trustworthy Machine Learning

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

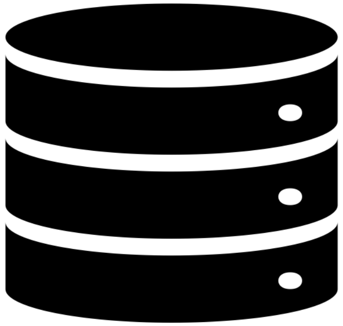
SprintML

Outline

I. Privacy Leakage in Machine Learning
   I. Adversary
   II. Treat-Space
II. Attribute Inversion Attacks
III. Model Inversion Attacks
IV. Membership Inference Attacks
   I. Shadow Models
   II. Loss-based Attacks
   III. Likelihood Ratio Attack
V. Intro to Differential Privacy
   I. Intuition
   II. Formula

Motivation: Extraction of Training Data

Training Data



train →

Diffusion model

generate →

($\ell_2$distance = 0.031)

Prompt: Ann Graham Lotz

Caption: Living in the light with Ann Graham Lotz

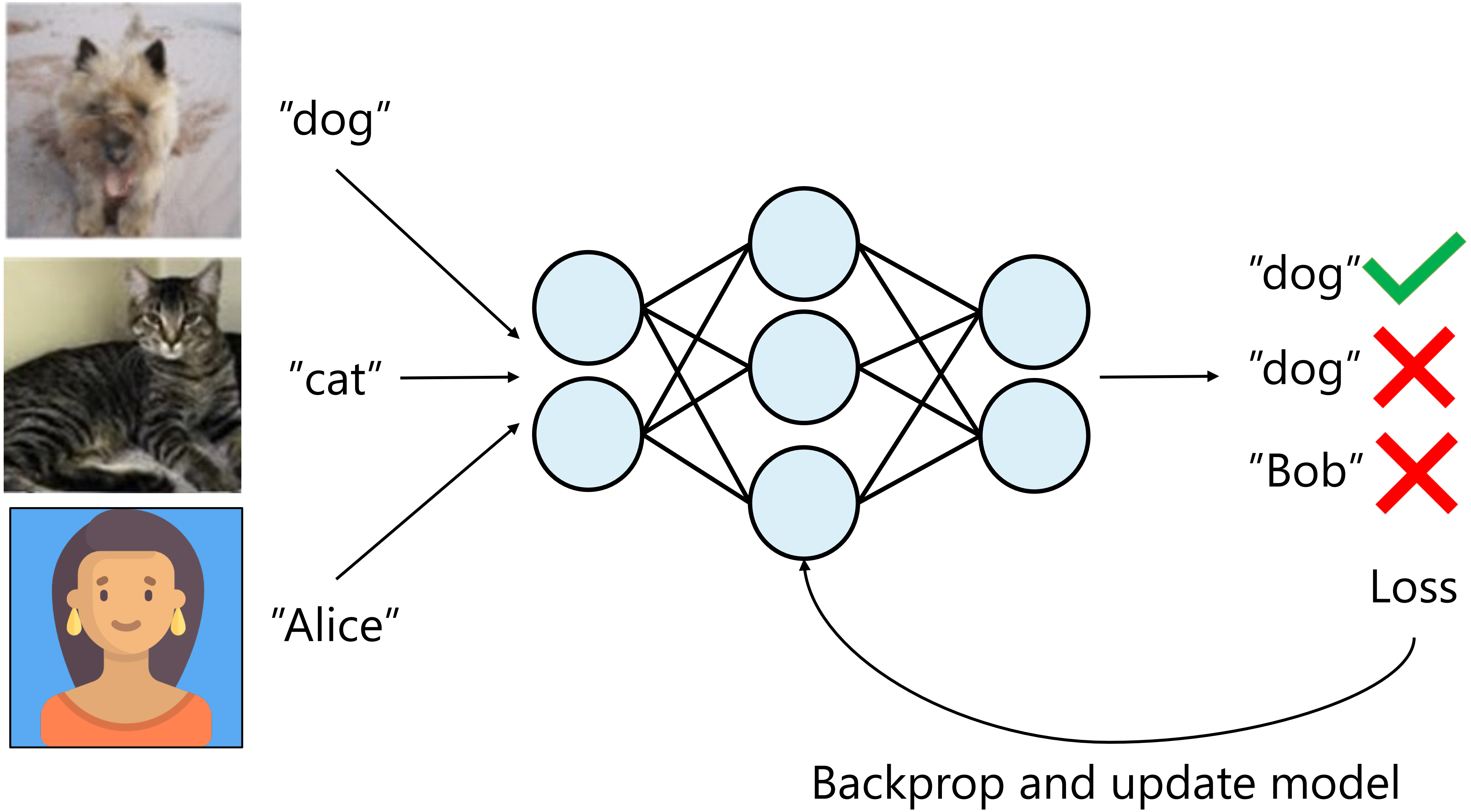
[*Wikipedia profile picture*](#)

Diffusion models memorize training images and emit them at test time.

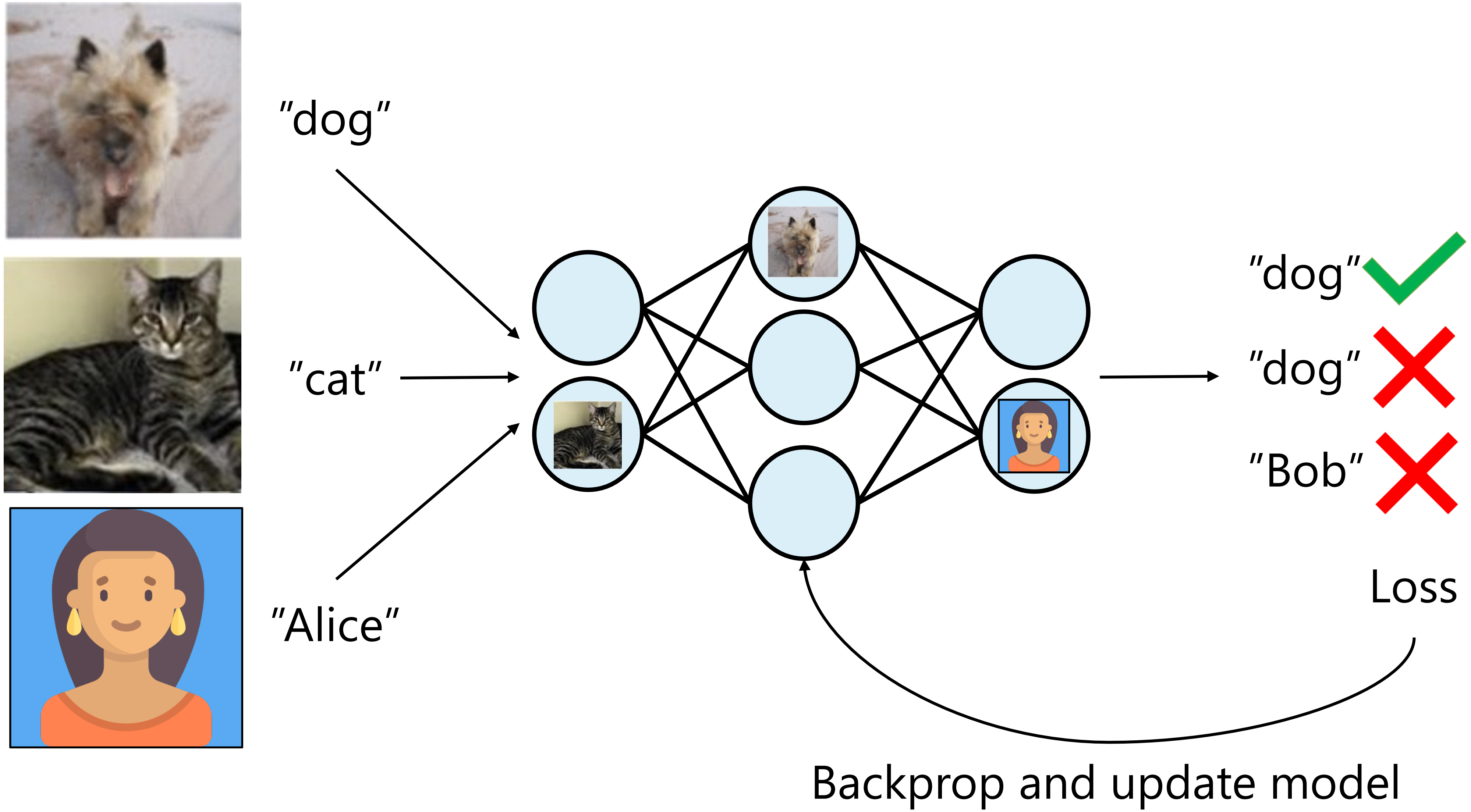
1. As the **quality of models increases so does privacy leakage**.
2. Extraction methodology:
   - Generate many examples using the diffusion model.
   - Perform **membership inference** to find training samples.

[Carlini et al., 2023]

Private data in model training

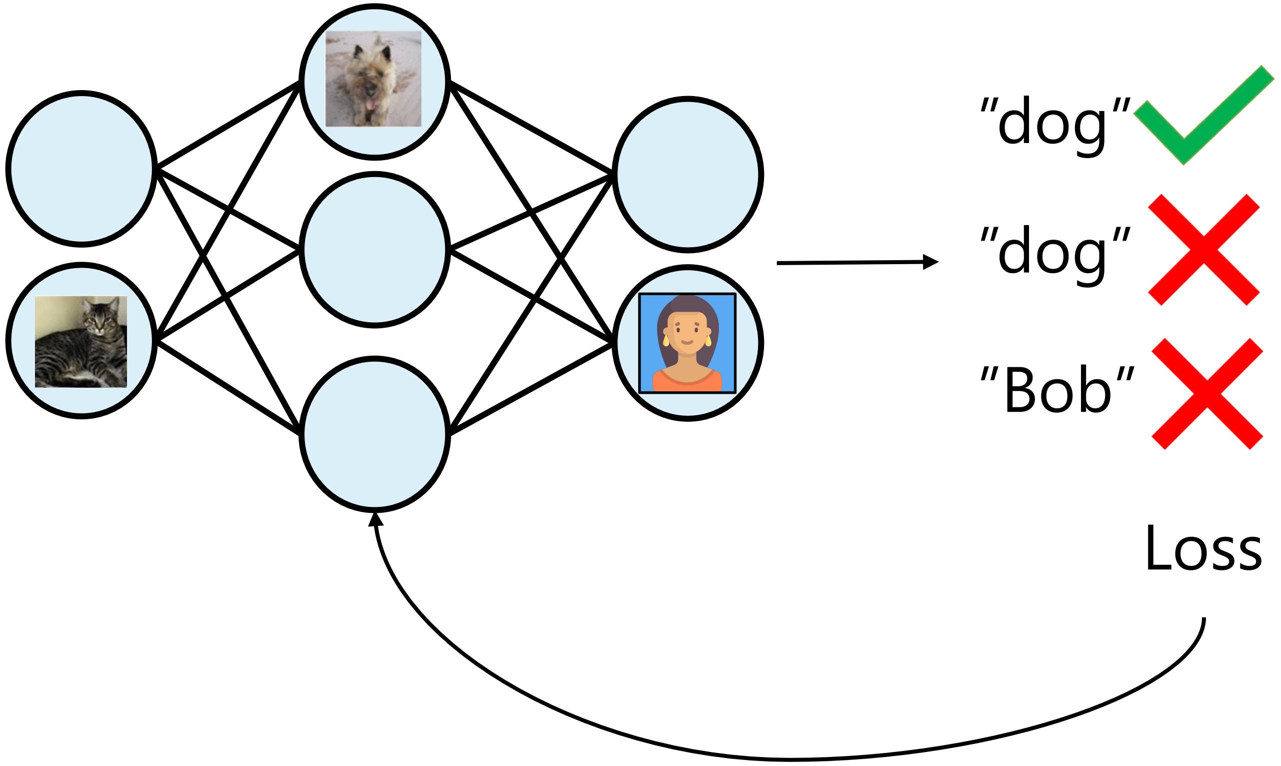


Private data in model training



Where can privacy leak?

From Parameters



From Predictions

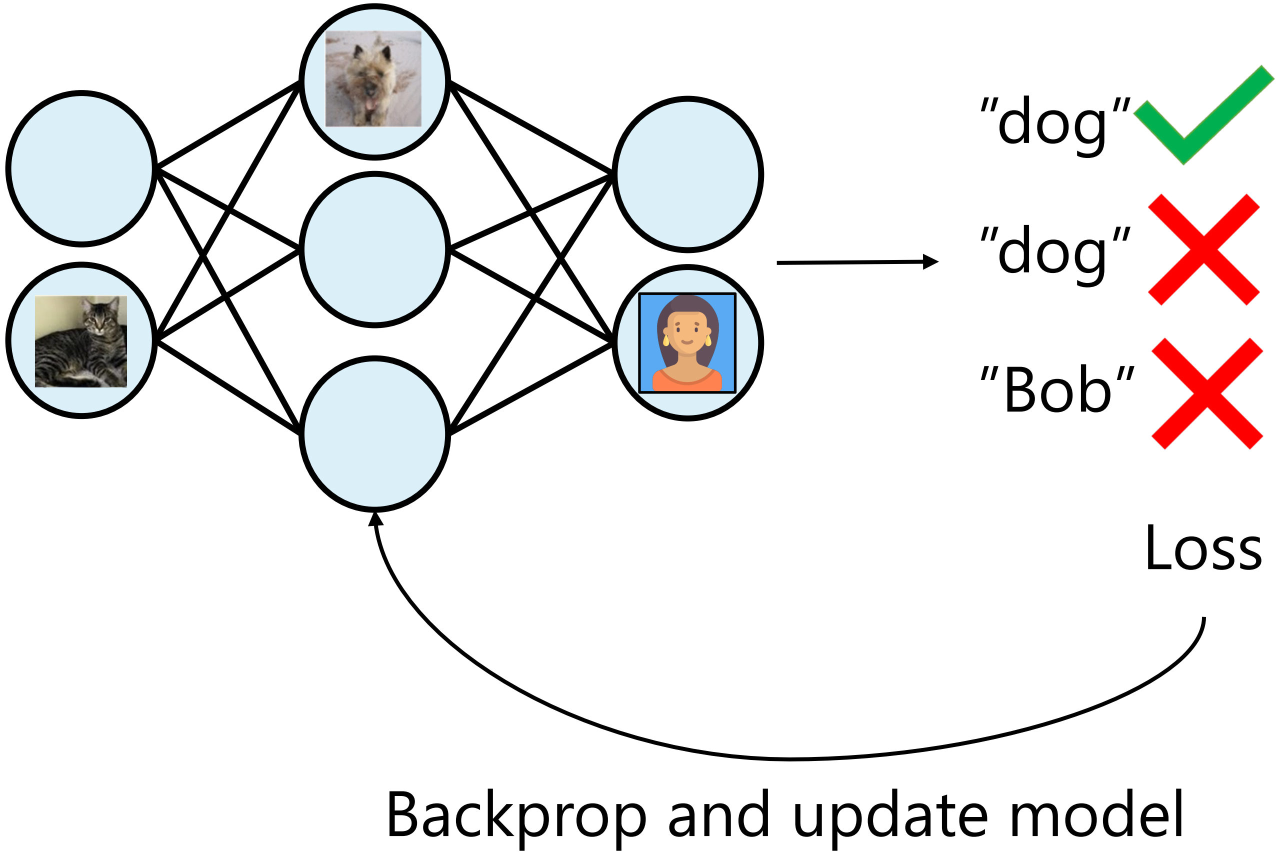
Backprop and update model

During Training

What are the adversary's abilities?

Observe/change the parameters

From Parameters



From Predictions

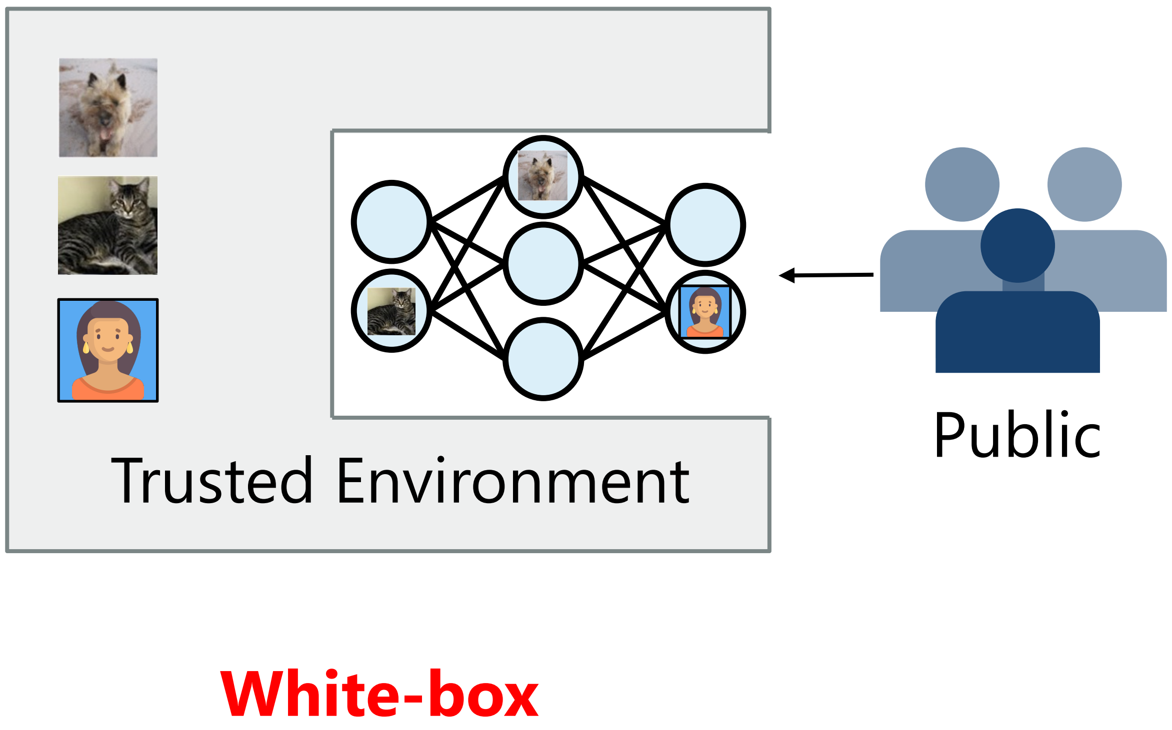
Knowledge about:

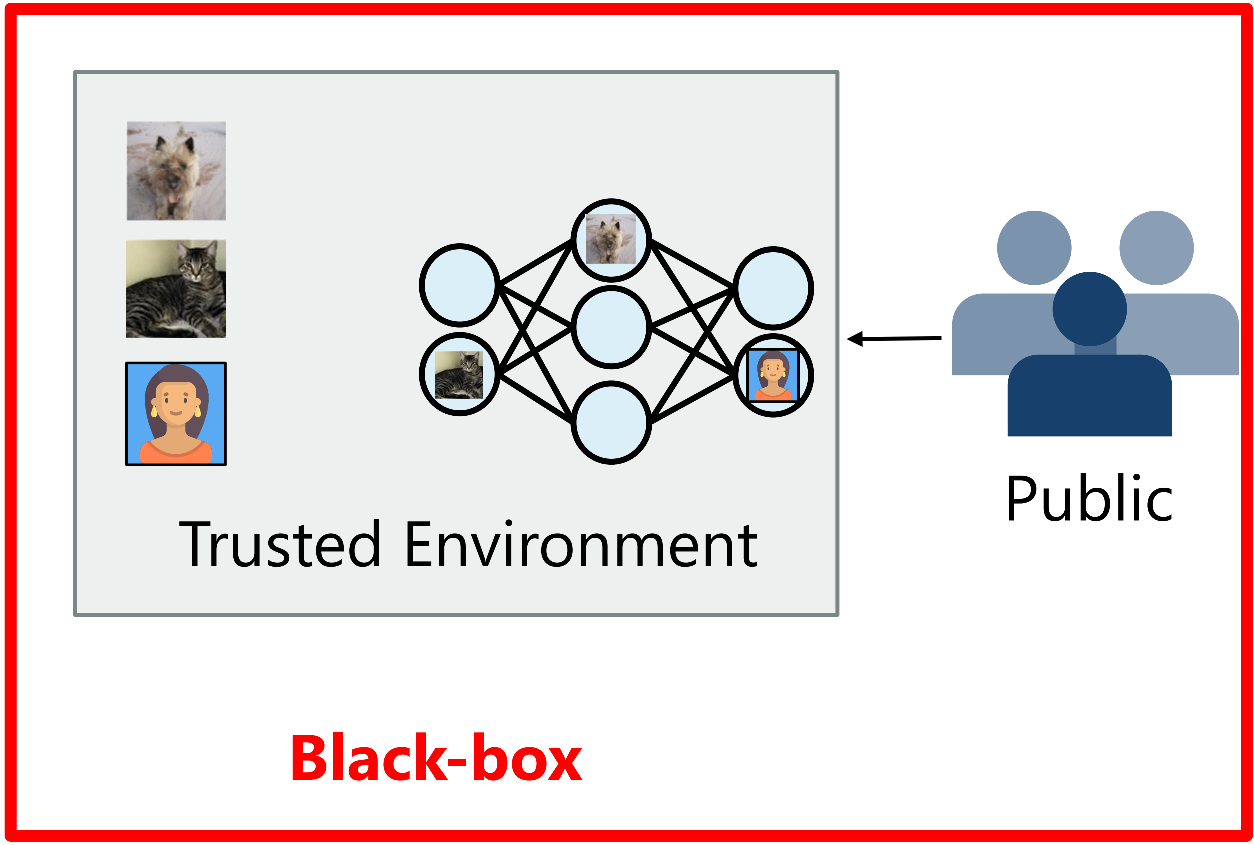
- **Model architecture**
- **Data attributes**
- **Data distribution**
- **Hyperparameters**

Observe/manipulate training During Training

[Papernot et al., 2018]

What is the threat space?





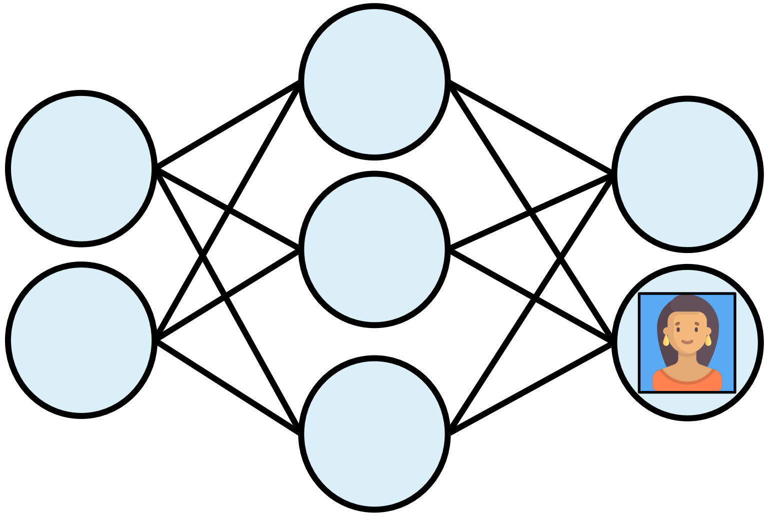
Attribute Inversion

Goal: Disclose the secret attribute of a training data point.

Name: Alice
Age: 34
Height: 1,72m
Smoker: Yes

Risk: **“High”**

Train



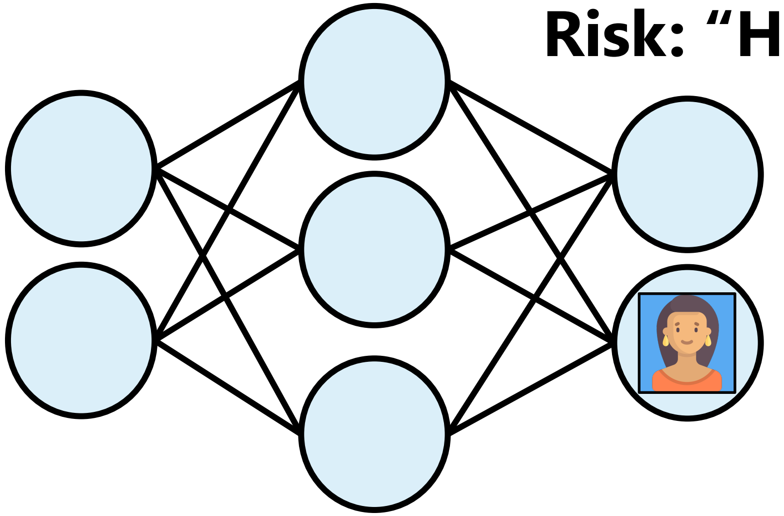
[Fredrikson et al., 2015]

Attribute Inversion

Goal: Disclose the secret attribute of a training data point.

Name:	Alice
Age:	34
Height:	1,72m
Smoker:	No

Predict



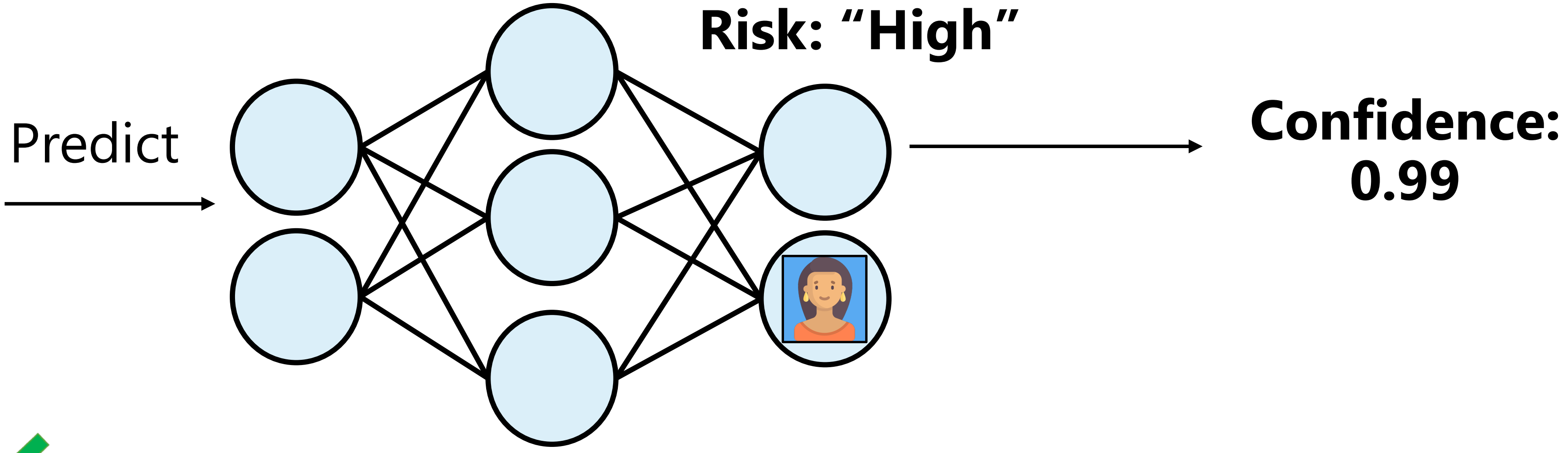
Risk: “High”

Confidence: 0.74

Attribute Inversion

Goal: Disclose the secret attribute of a training data point.

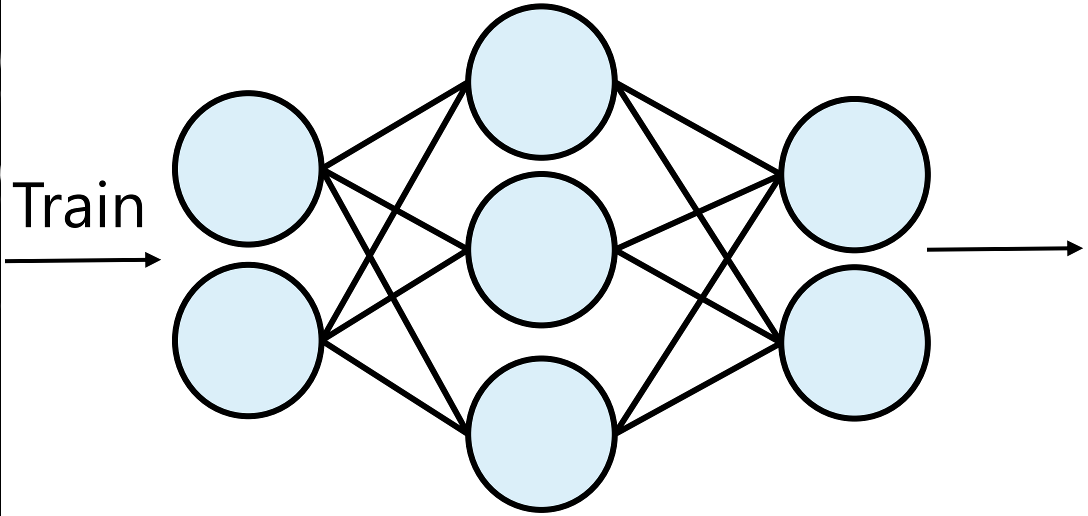
Name:	Alice
Age:	34
Height:	1,72m
Smoker:	Yes

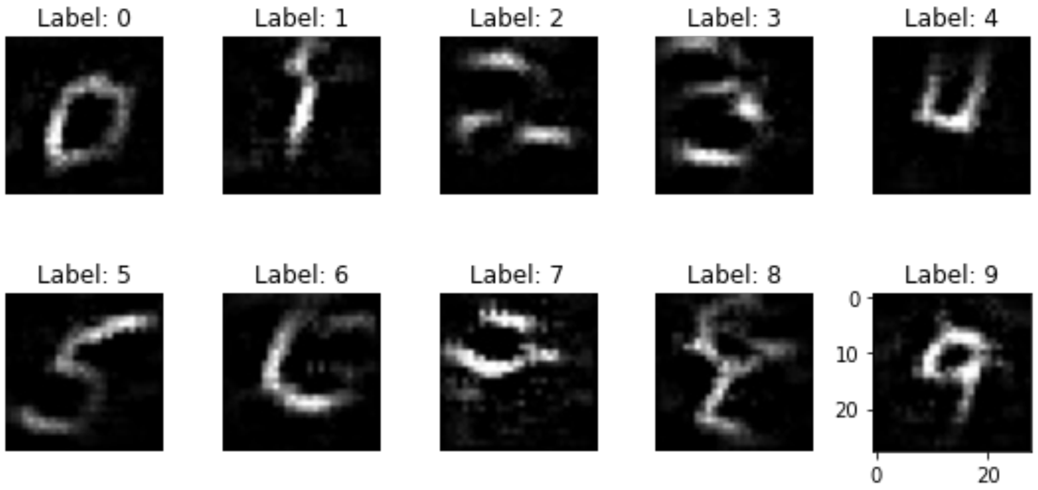


Model Inversion

Goal: Disclose a “prototype” of each training class.

MNIST

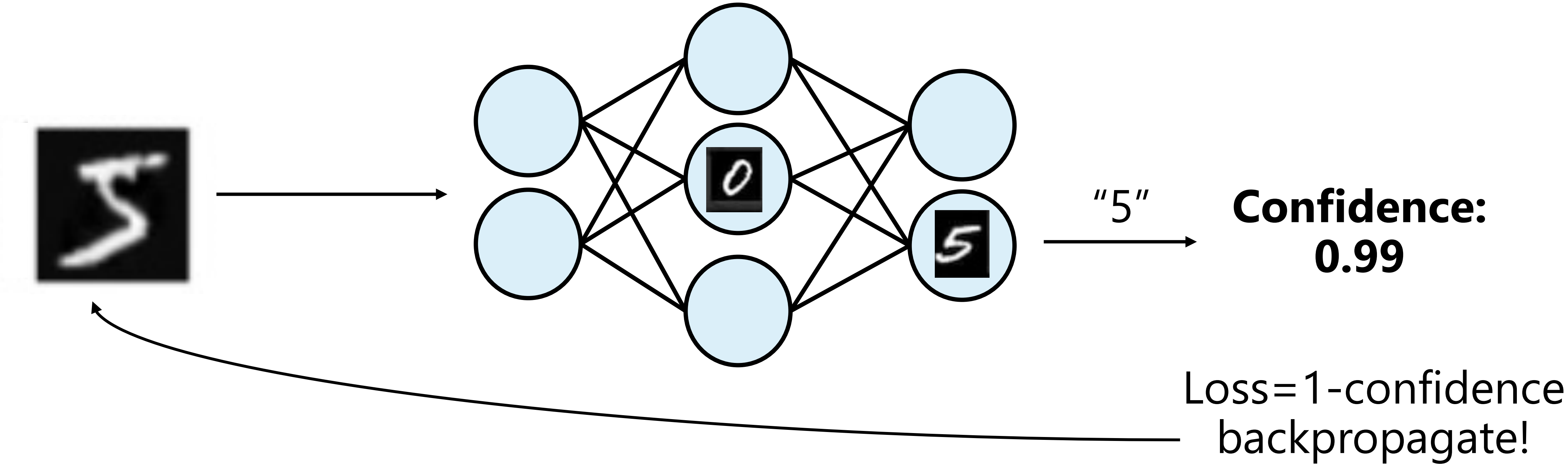




[Fredrikson et al., 2015]

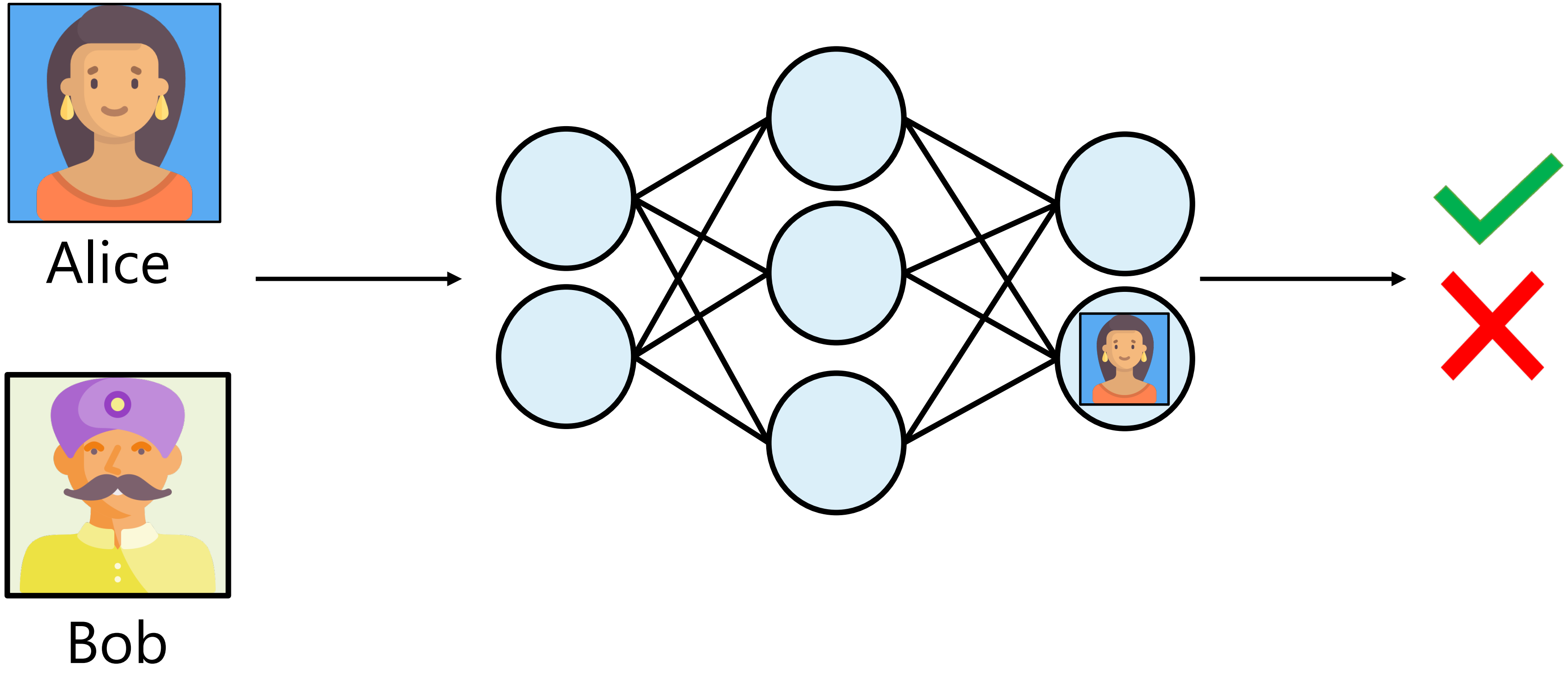
Model Inversion

Goal: Disclose a “prototype” of each training class.



Membership Inference Attacks (MIA)

Goal: Disclose whether a given data point was used to train the model.



[Shokri et al., 2017]

The Membership Inference "Game"

Challenger C

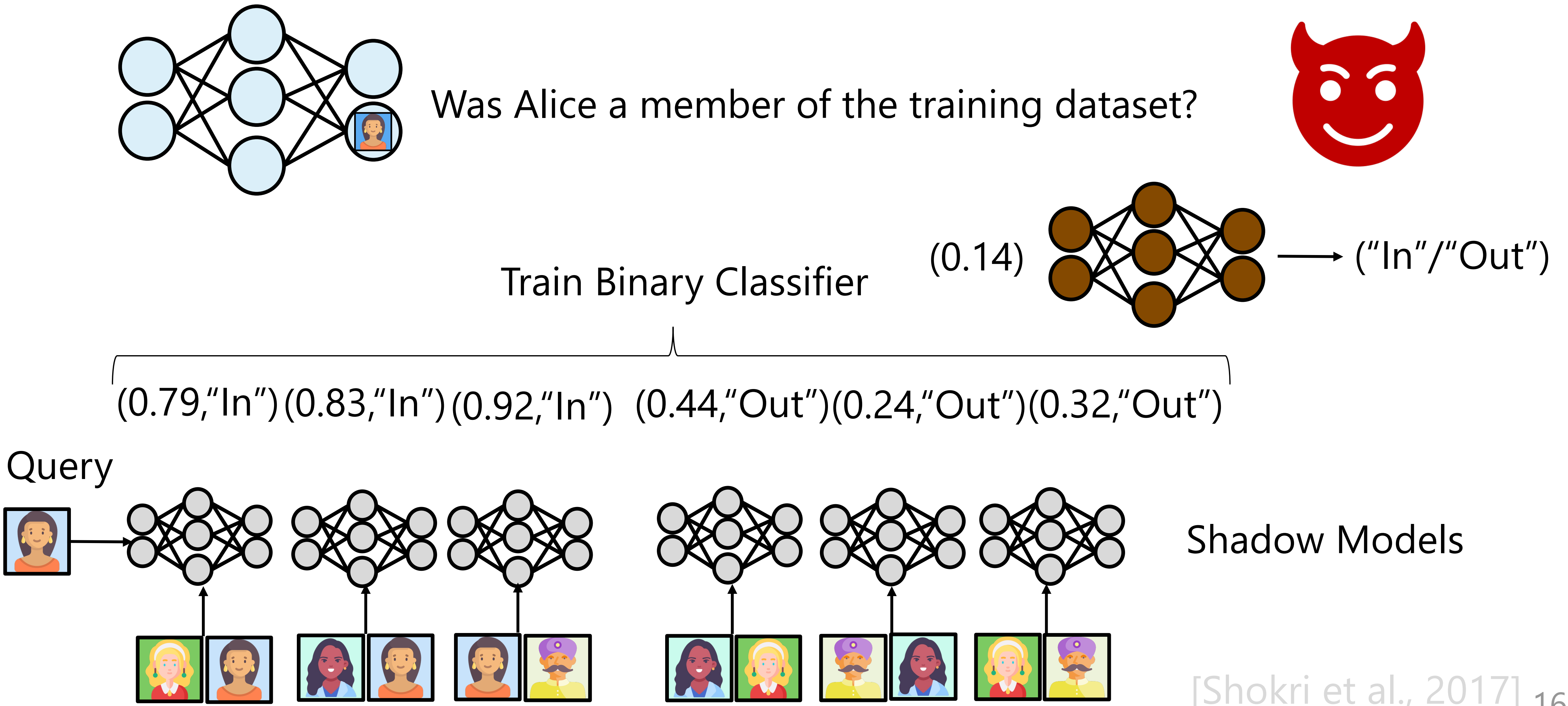
Adversary A

C samples training dataset $D \leftarrow \mathcal{D}$ and trains model $f \leftarrow \mathcal{A}(D)$ with algorithm $\mathcal{A}$.

1. C flips a coin b, and samples a point $(x, y) \in \mathcal{D} \backslash \mathrm{D}$ if $b = 0$. Otherwise, if $b = 1$, sample $(x, y) \in \mathrm{D}$.
2. C sends (x, y) to A.
3. A gets access to $\mathcal{D}$ and model f and outputs $\hat{b}$.
4. A wins if $\hat{b} = b$.

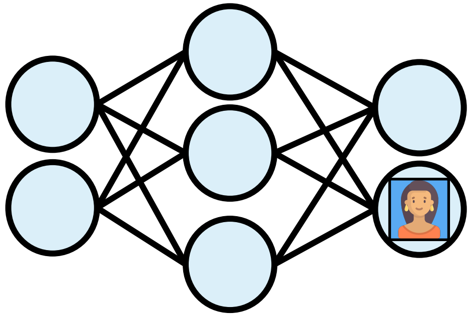
[Carlini et al., 2018]

Shadow-Model Based Membership Inference

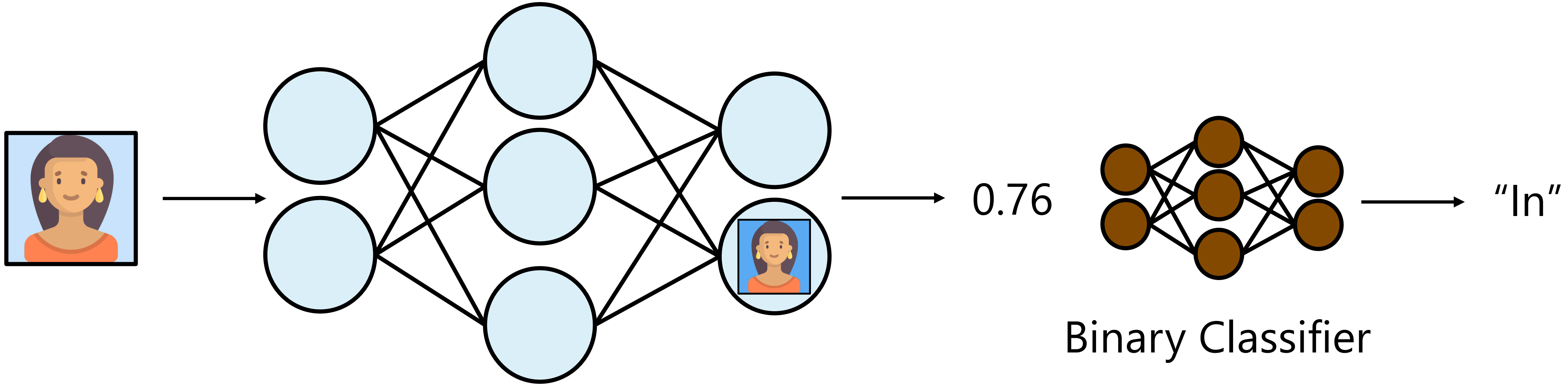


[Shokri et al., 2017]

Shadow-Model Based Membership Inference



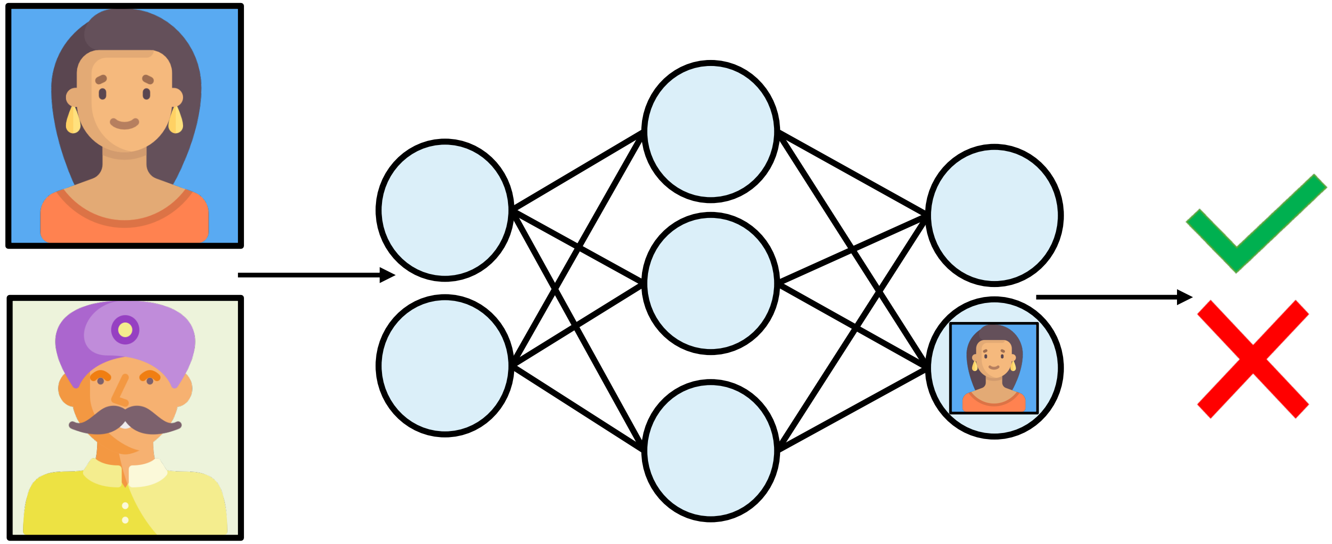
Was Alice a member of the training dataset?

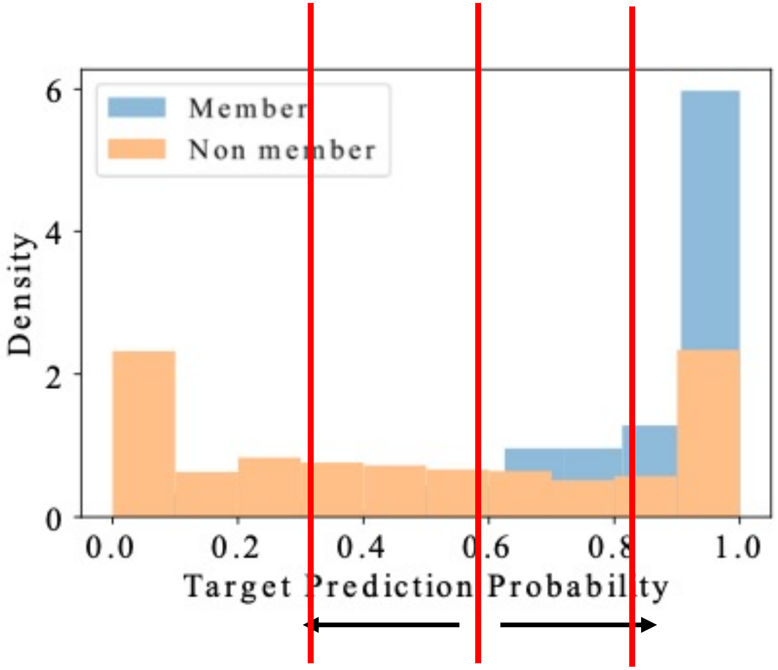


Target Model

[Shokri et al., 2017]

Threshold-based Membership Inference



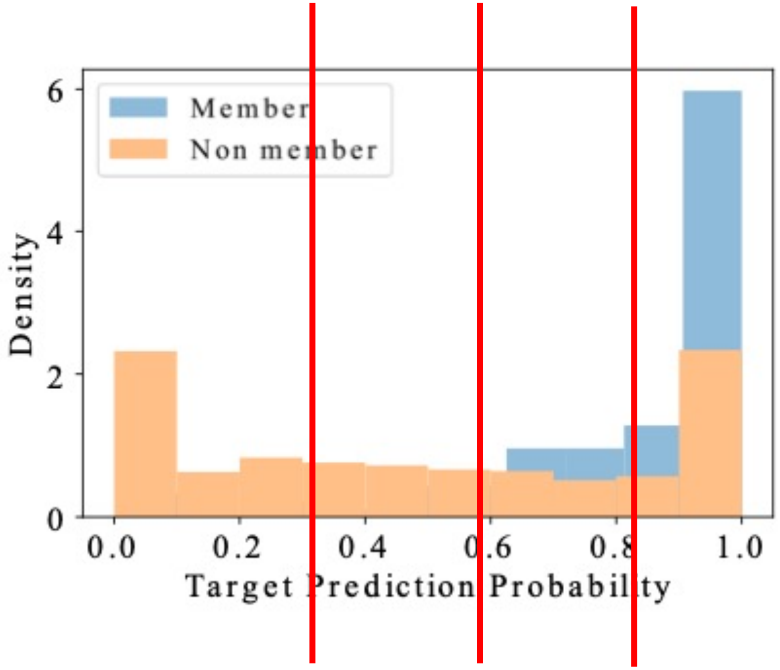


Non-Members Members

[Yeom et al., 2018]

Attacker's Prediction Success

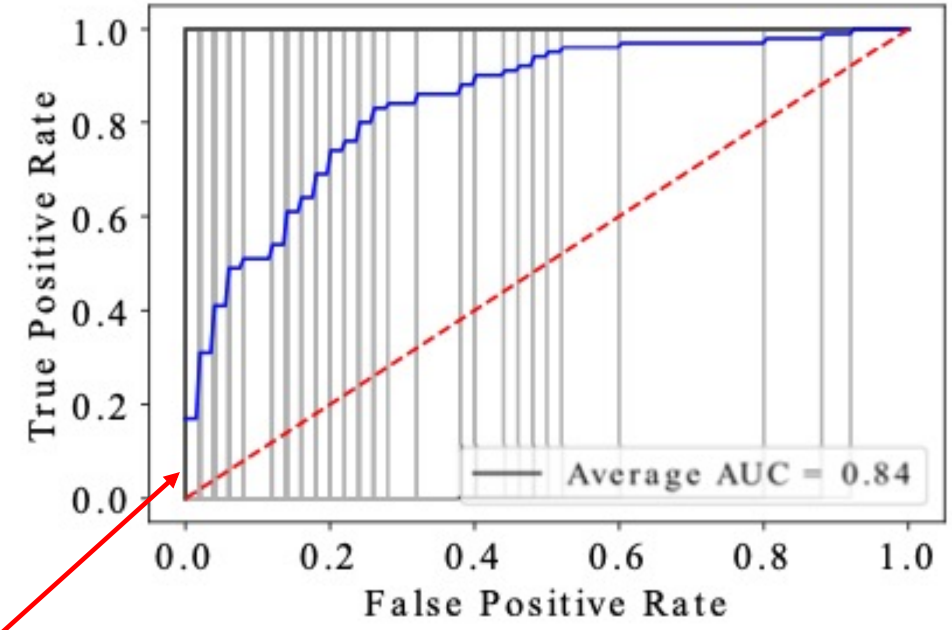
FPR=, TRP=



FPR=, TRP=

FPR=, TRP=

Receiver operating characteristic curve
(ROC Curve)



Important!
TPR at low FPR

AUC: Area Under the Curve
Average ☹

FPR: False Positive Rate: Non-member is classified as member.

TPR: True Positive Rate: Member is classified as member.

Likelihood Ratio Attack (LiRA)

Consider two distributions over models:

$\mathbb{Q}_{in} = \{f \leftarrow \mathcal{A}(D \cup \{(x, y)\}) \mid D \leftarrow \mathcal{D}\}$ ← Models trained with data point (x, y)

$\mathbb{Q}_{out} = \{f \leftarrow \mathcal{A}(D \setminus \{(x, y)\}) \mid D \leftarrow \mathcal{D}\}$ ← Models trained without data point (x, y)

Thresholding the Likelihood-ratio Test between the two hypotheses:

$$\Lambda(f; x, y) = \frac{p(f \mid \mathbb{Q}_{in}(x,y))}{p(f \mid \mathbb{Q}_{out}(x,y))}$$

Intractable!

p probability density function of f under distribution of model parameters

Simplify by using loss instead:

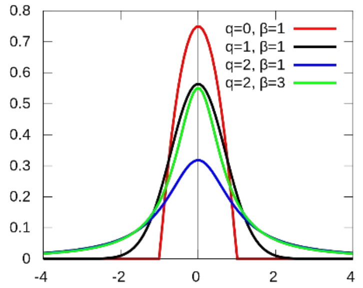
$\mathbb{Q}'_{in\backslash out}$ distribution of losses on (x, y)

$$\Lambda(f; x, y) = \frac{p(\ell(f(x),y) \mid \mathbb{Q}'_{in}(x,y))}{p(\ell(f(x),y) \mid \mathbb{Q}'_{out}(x,y))}$$

[Carlini et al., 2022]

Likelihood Ratio Attack

Train shadow models to estimate $\mathbb{Q}'_{in}$ and $\mathbb{Q}'_{out}$.
Simplify by assumption that they follow Gaussian distribution

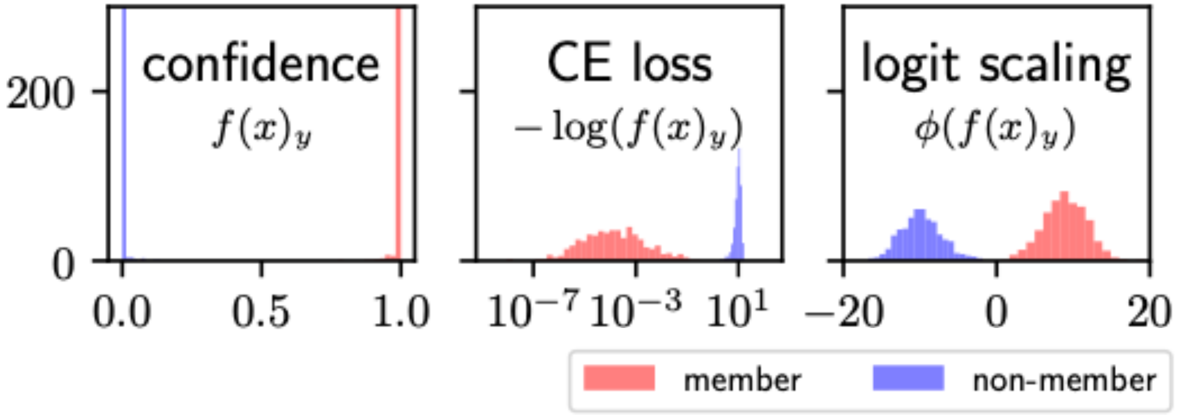


Four variables to be estimated:

means: μ_{in}, μ_{out}

standard deviations: $\sigma_{in}, \sigma_{out}$

How to ensure losses are indeed Gaussians?



Logit scaling to the model confidence:
$\phi(p) = \log \frac{p}{1-p}$, for $p = f(x)_y$

[Carlini et al., 2022]

Likelihood Ratio Attack

Require: model f, example (x, y), data distribution $\mathbb{D}$

```
confs_in = {}
confs_out = {}
for N times do
    D_attack ←$ 𝔻                                   ▷ Sample a shadow dataset
    f_in ← T(D_attack ∪ {(x, y)})                   ▷ train IN model
    confs_in ← confs_in ∪ {φ(f_in(x)_y)}
    f_out ← T(D_attack \ {(x, y)})                  ▷ train OUT model
    confs_out ← confs_out ∪ {φ(f_out(x)_y)}
end for
μ_in ← mean(confs_in)
μ_out ← mean(confs_out)
σ²_in ← var(confs_in)
σ²_out ← var(confs_out)
conf_obs = φ(f(x)_y)                                ▷ query target model
return Λ = p(conf_obs | N(μ_in, σ²_in)) / p(conf_obs | N(μ_out, σ²_out))
```

$$\Lambda = \frac{p(\mathrm{conf}_{\mathrm{obs}} \mid \mathcal{N}(\mu_{\mathrm{in}}, \sigma^2_{\mathrm{in}}))}{p(\mathrm{conf}_{\mathrm{obs}} \mid \mathcal{N}(\mu_{\mathrm{out}}, \sigma^2_{\mathrm{out}}))}$$

Effective but computationally costly

How to defend against MIA?

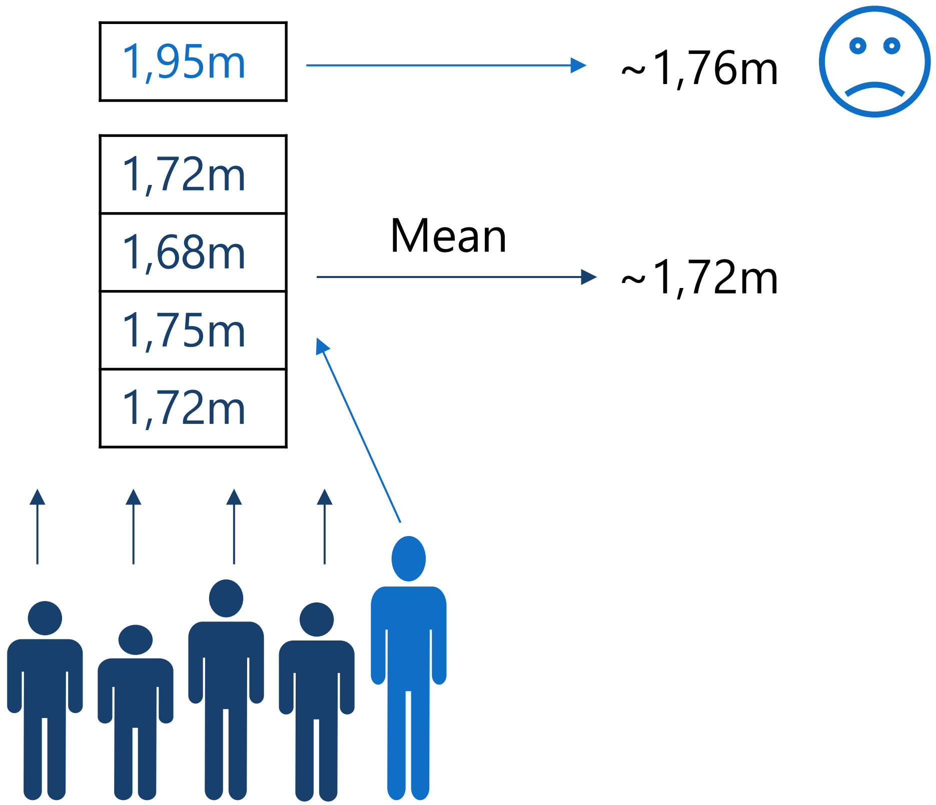
1. Add noise to confidence vector
2. Do not output prediction probability, just output labels

3. Reduce overfitting: Regularization, different losses

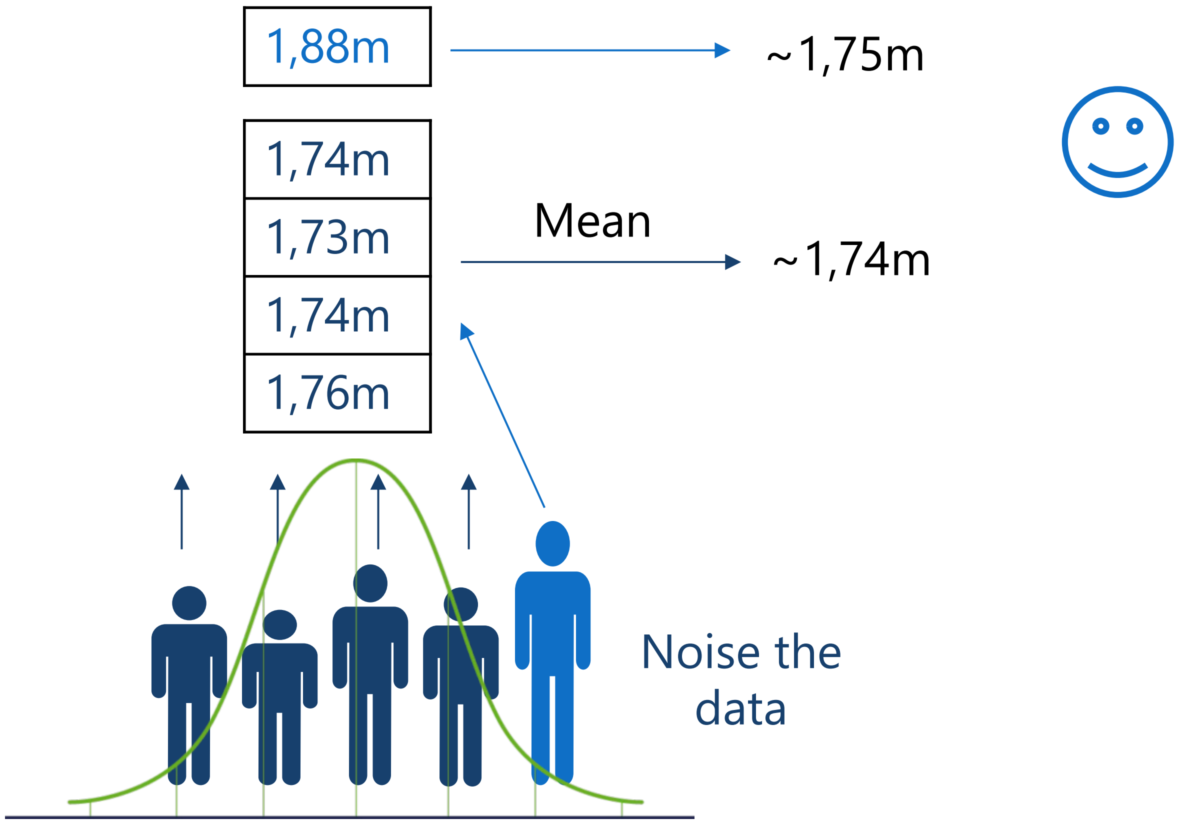
Empirical!

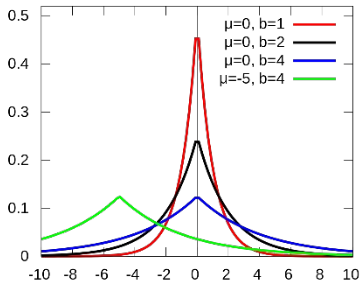
Can we get guarantees?

Differential Privacy – Example 1

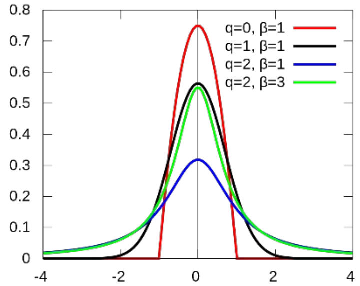


Differential Privacy – Example 1



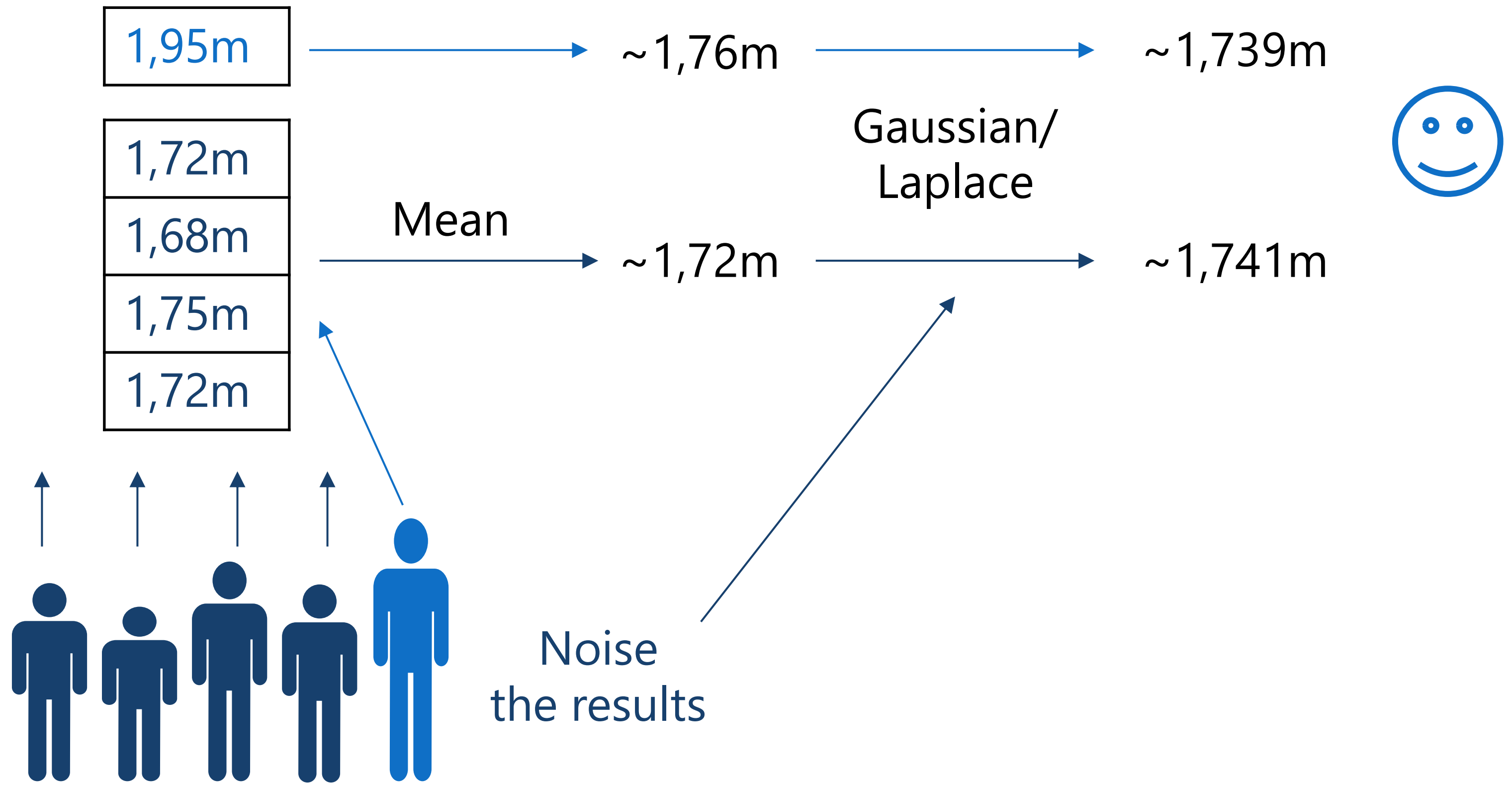


Laplace

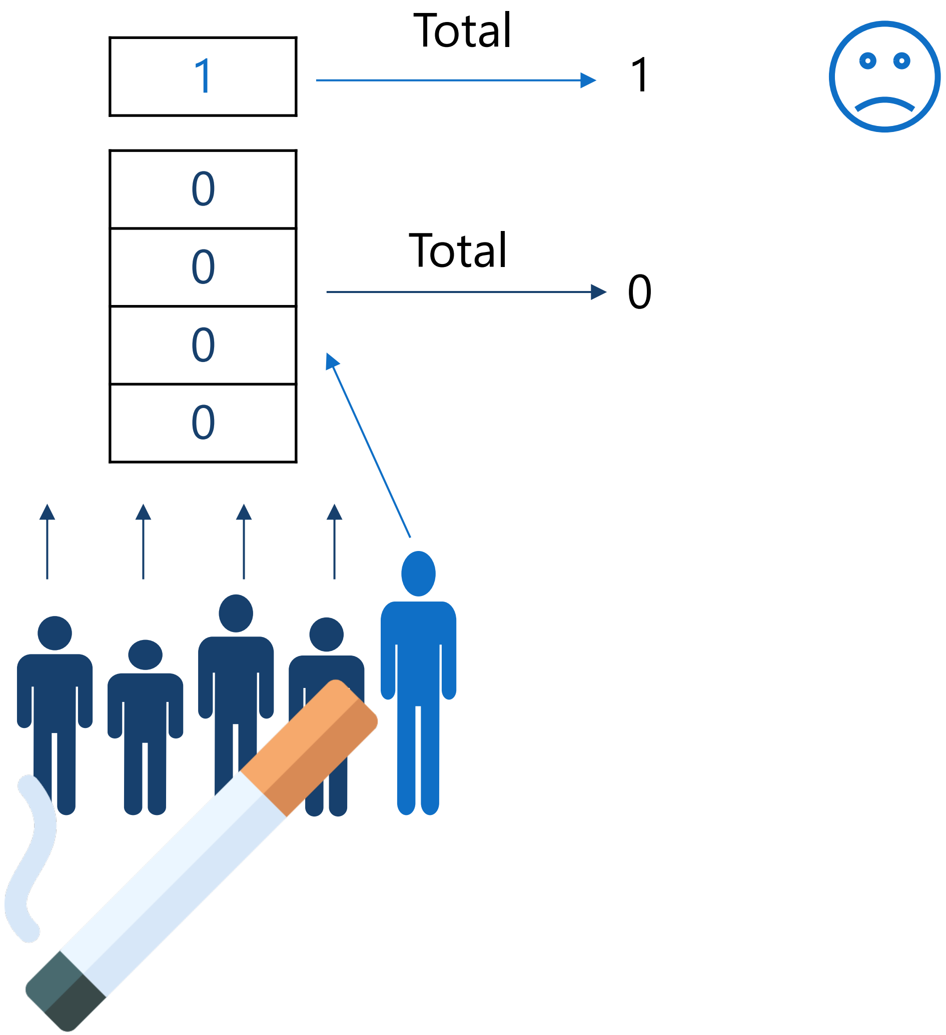


Gauss

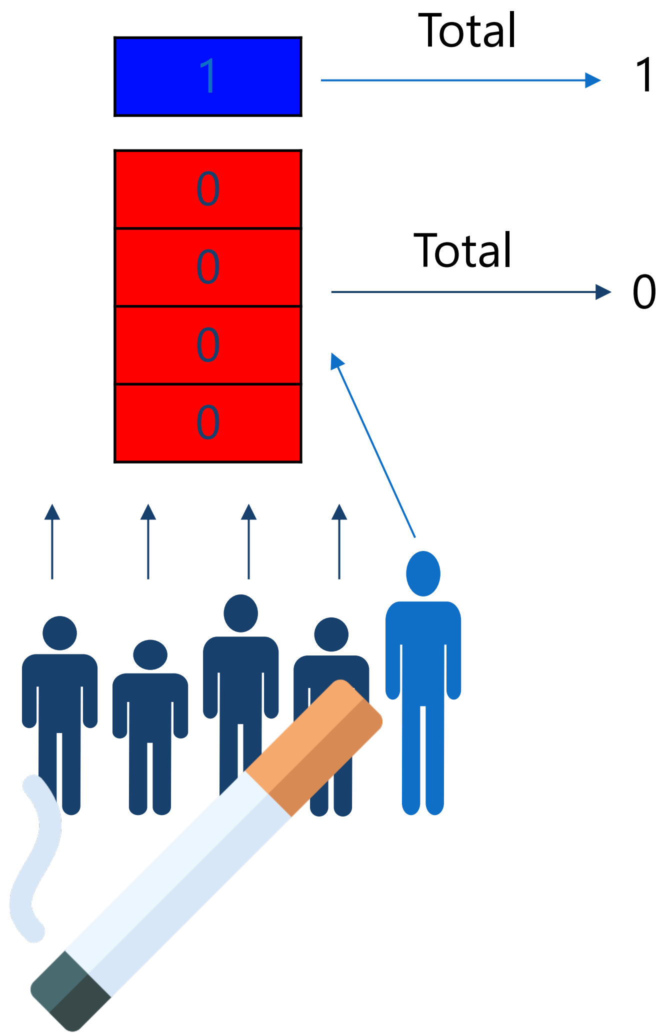
Differential Privacy – Example 1

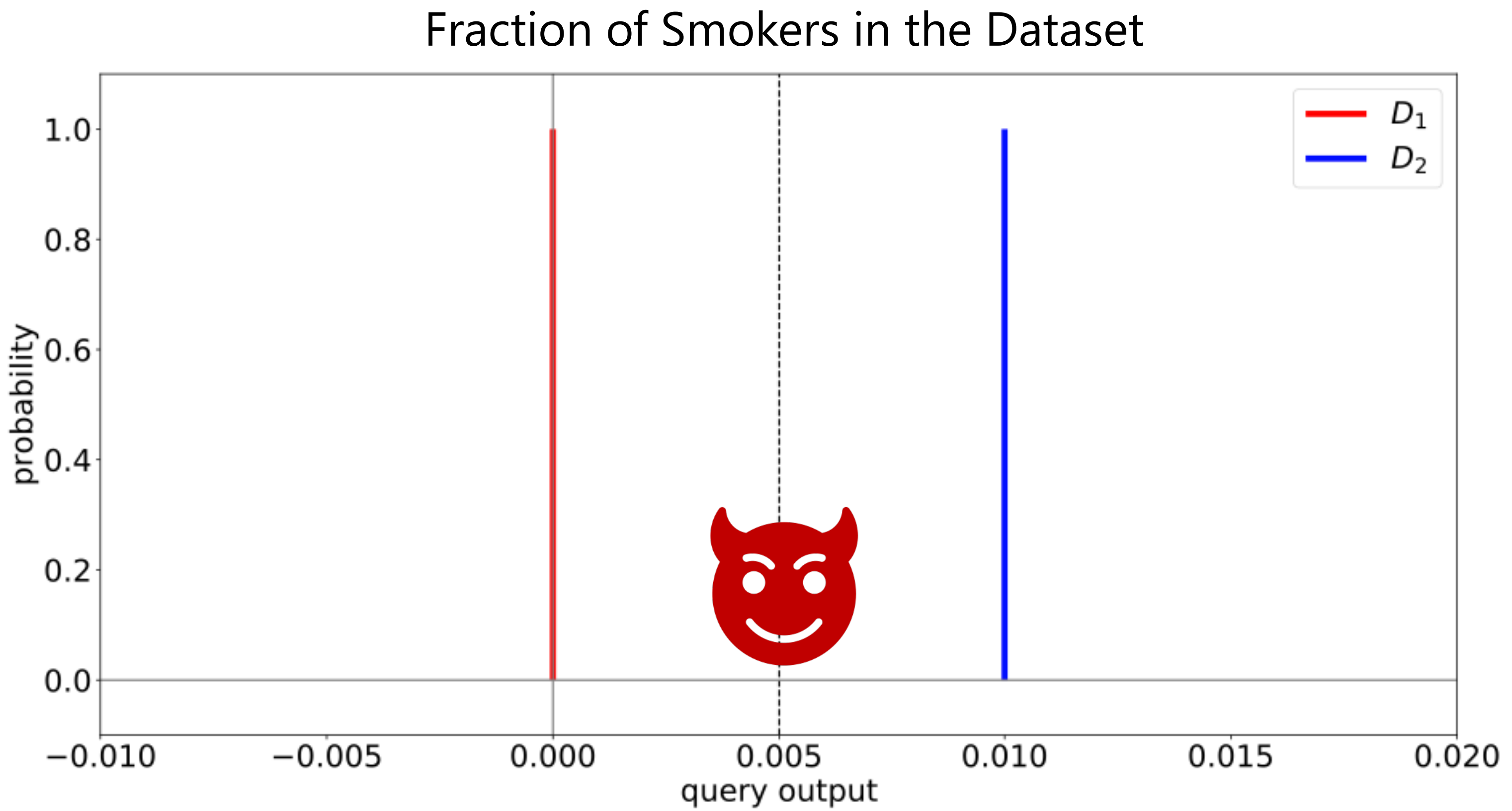


Differential Privacy – Example 2



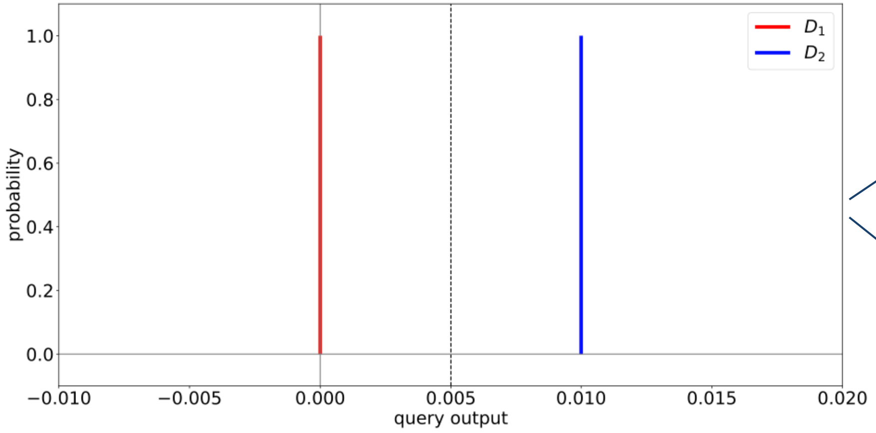
Deterministic algorithms yield no privacy



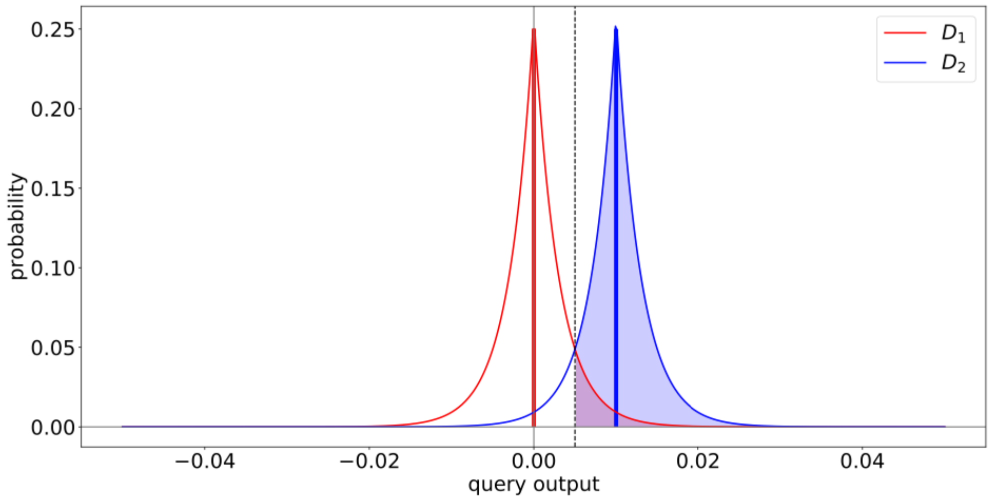


How much noise to add?

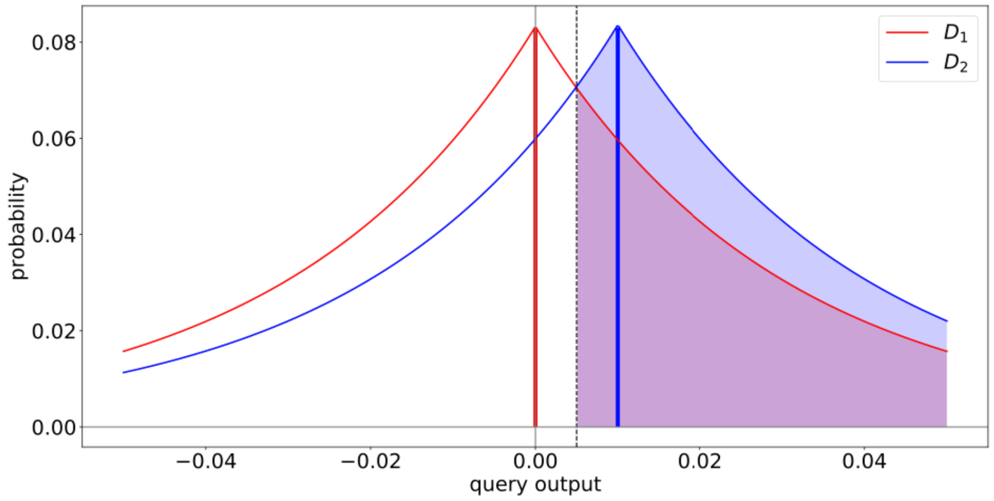
Fraction of Smokers in the Dataset



Little Noise



Much Noise



Formalizing Differential Privacy

Intuition: An algorithm $\boldsymbol{M}$ provides $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$-Differential Privacy (DP) if it produces "roughly same" outputs on any pair of training datasets $\boldsymbol{d}$ and $\boldsymbol{d'}$ that differ only by a single data point.

Closeness

Probability of the Closeness Violation

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

Randomized Algorithm

Possible Outputs

[Dwork, 2006]

Formalizing Differential Privacy

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

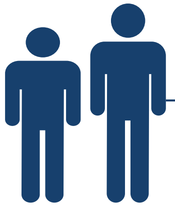
$\boldsymbol{\varepsilon} > 0$: Privacy budget
Smaller → more privacy

$\boldsymbol{\delta} \in [0,1]$: Probability of violating closeness
Smaller → more privacy, usually chosen $< 1/n$ with n data points

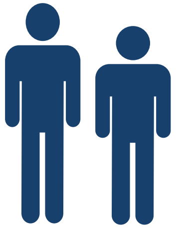
With $(\varepsilon, 0)$, we fulfill pure ε-DP, with $\delta > 0$, we have approximate DP.

Properties Differential Privacy

Parallel Composition: If $\boldsymbol{M}$(x) fulfills ε, δ-DP, and if we split our data $\mathcal{D}$ into k disjoint subsets $\mathcal{D} = x_1 \cup \cdots \cup x_k$, then the mechanism that releases all results $M(x_1), \ldots, \boldsymbol{M}(x_k)$ is ε, δ-DP.

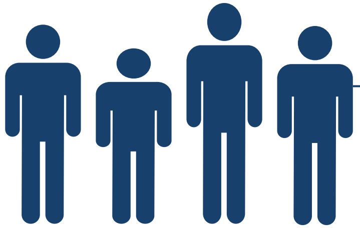


1,75m
1,72m



1,72m
1,68m

Sequential Composition: If $\boldsymbol{M}_1$(x) fulfills ε_1, δ_1-DP and $\boldsymbol{M}_2$(x) fulfills ε_2, δ_2-DP, then **G**(x) = ($\boldsymbol{M}_1$(x), $\boldsymbol{M}_2$(x)) fulfills ($\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2$)-DP.

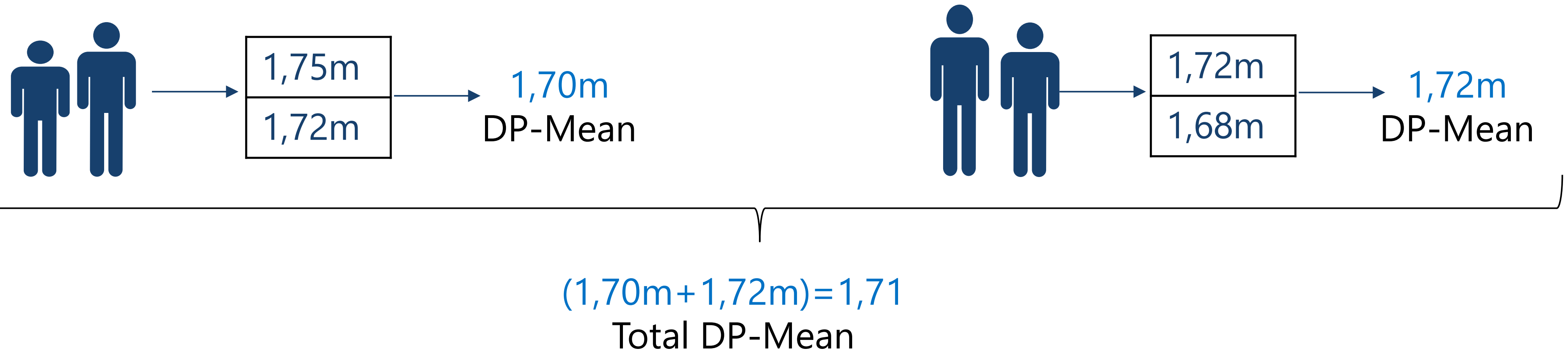


1,72m
1,68m
1,75m
1,72m

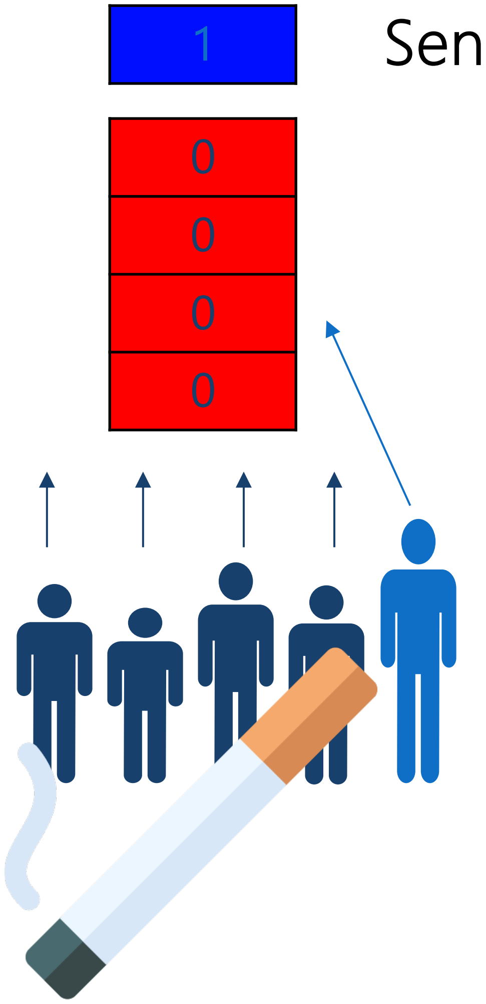
0
0
0
0

Properties Differential Privacy

Postprocessing guarantees: If an output of an (ε, δ)-DP mechanism is further processed or transformed, the guarantees remain.



How to find the noise level: Sensitivity



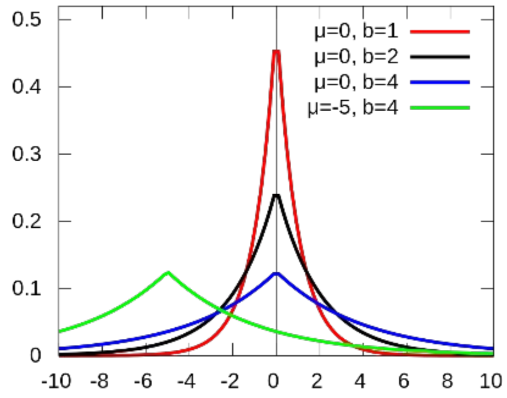
Sensitivity: By how much can a single data point change the outcome.

Sensitivity Δf of a function f operating on the neighboring datasets d and d' is defined as $\Delta \mathrm{f} = \max(\|\mathrm{f}(\mathrm{d}) - \mathrm{f}(\mathrm{d}')\|)$.

Sensitivity of any counting function is 1.

We can use different norms to calculate the sensitivity.

Laplace Mechanism



Laplace

Given a function $f\colon \mathcal{D} \rightarrow \mathbb{R}^d$ where $\mathcal{D}$ is the domain of the dataset and d is the dimension of the output, the Laplace mechanism adds Laplace noise to the output of f.

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

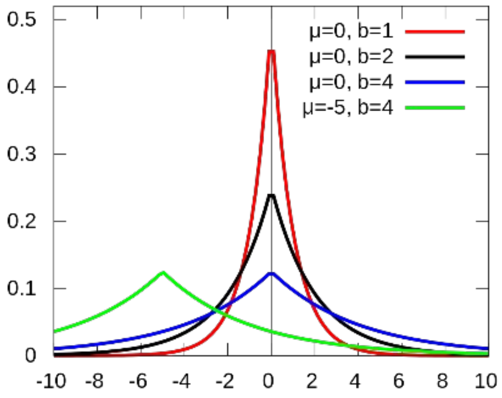
Laplace (noise) Distribution

b is the scale parameter of the Laplace distribution.

The Laplace mechanism is $M(D) = f(D) + Lap(0|b)^d$.

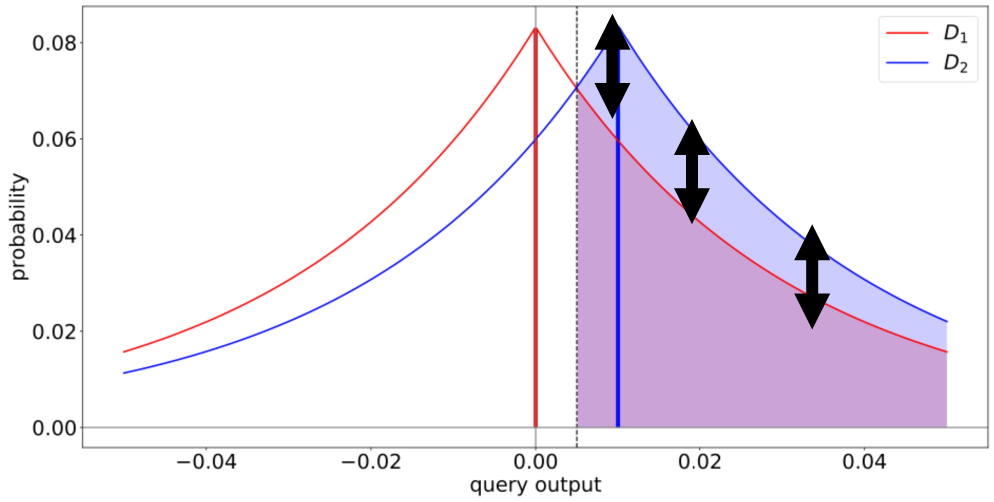
If we choose $b = \frac{\Delta f}{\varepsilon}$, this mechanism fulfills ε-DP.

Proof Sketch



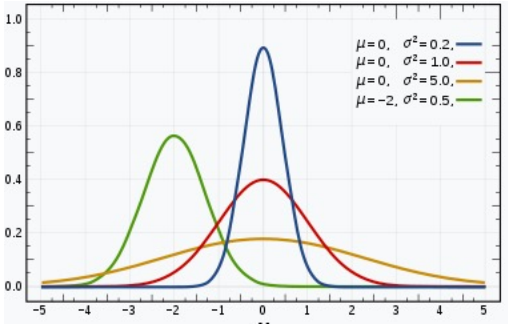
Laplace

If we choose $b = \frac{\Delta f}{\varepsilon}$, this mechanism fulfills ε-DP.



Show that $\frac{\Pr[\boldsymbol{M}(d) \in S]}{\Pr[\boldsymbol{M}(d') \in S]} \leq e^{\boldsymbol{\varepsilon}}$.

Gaussian Mechanism



Gauss

Given a function $f\colon \mathcal{D} \to \mathbb{R}^d$ where $\mathcal{D}$ is the domain of the dataset and d is the dimension of the output, the Gaussian mechanism adds Gaussian noise to the output of f.

$$\mathcal{N}(x|\sigma,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Gaussian (noise) Distribution

σ is the standard deviation, and μ the mean.

The Gaussian mechanism is $M(D) = f(D) + \mathcal{N}(0|\sigma,\mu)^d$.

If we choose $\mu = 0$ and $\sigma^2 = \frac{2\ln\left(\frac{1.25}{\delta}\right)(\Delta f)^2}{\varepsilon^2}$, this mechanism fulfills (ε,δ)-DP.

Laplace vs Gaussian Mechanism

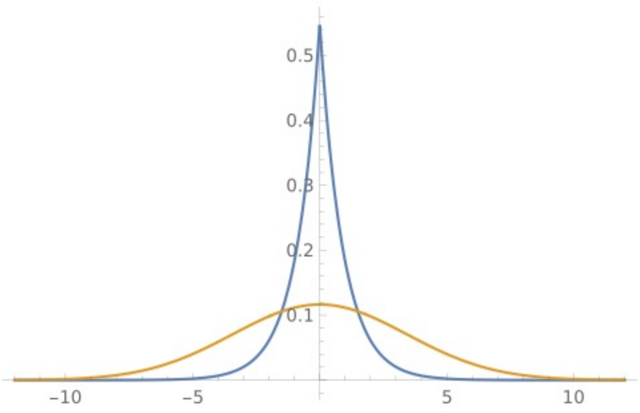
$$Lap(x|b,\mu) = \frac{1}{2b} e^{\boxed{-\frac{|x-\mu|}{b}}} \qquad \mathcal{N}(x|\sigma,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{\boxed{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}$$

ε-DP vs. (ε, δ)-DP

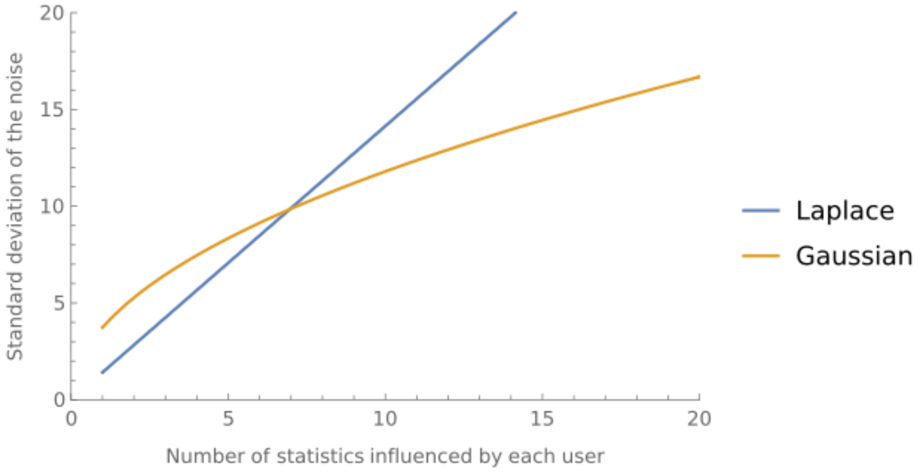
Privacy Guarantees

$L1$ vs $L2$

Norms



Noise distributions at the same $\varepsilon \approx 1$



Standard deviation after sequential execution

Further Reading

[1] Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. P. (2018, April). Sok: Security and privacy in machine learning. In *2018 IEEE European symposium on security and privacy (EuroS&P)* (pp. 399-414). IEEE.
[2] Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322-1333. 2015.
[3] Shokri, Reza, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models." In *2017 IEEE symposium on security and privacy (SP)*, pp. 3-18. IEEE, 2017.
[4] Yeom, Samuel, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. "Privacy risk in machine learning: Analyzing the connection to overfitting." In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268-282. IEEE, 2018.
[5] Choquette-Choo, Christopher A., Florian Tramer, Nicholas Carlini, and Nicolas Papernot. "Label-only membership inference attacks." In *International conference on machine learning*, pp. 1964-1974. PMLR, 2021.
[6] Carlini, Nicholas, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. "Membership inference attacks from first principles." In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897-1914. IEEE, 2022.
[7] Dwork, Cynthia. "Differential privacy." In *International colloquium on automata, languages, and programming*, pp. 1-12. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
[8] Carlini, Nicolas, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. "Extracting training data from diffusion models." In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253-5270. 2023.

Thank you!

Franziska Boenisch and Adam Dziedzic
boenisch@cispa.de, adam.dziedzic@cispa.de
sprintml.com
Course on Trustworthy Machine Learning