

Privacy and Security in Machine Learning

Mainack Mondal

CS 60081
Autumn 2024



Roadmap

- Attacks on machine learning models
 - “Data Privacy in Machine Learning” by Reza Shokri, NUS
- Privacy preserving machine learning
 - Course taught by Aurélien Bellet
 - http://researchers.lille.inria.fr/abellet/teaching/private_machine_learning_course.html

Data Privacy in Machine Learning

Reza Shokri

Data Privacy and Trustworthy ML Research Lab
National University of Singapore



reza@comp.nus.edu.sg



@rzshokri

Threats to Data Privacy

Threats to Data Privacy

- Unauthorized access to data, and data breaches
- Massive data collection

Threats to Data Privacy

Direct and intentional leakage

- Unauthorized access to data, and data breaches
- Massive data collection

Threats to Data Privacy

Direct and intentional leakage

- Unauthorized access to data, and data breaches
- Massive data collection

Indirect and unintentional leakage

Threats to Data Privacy

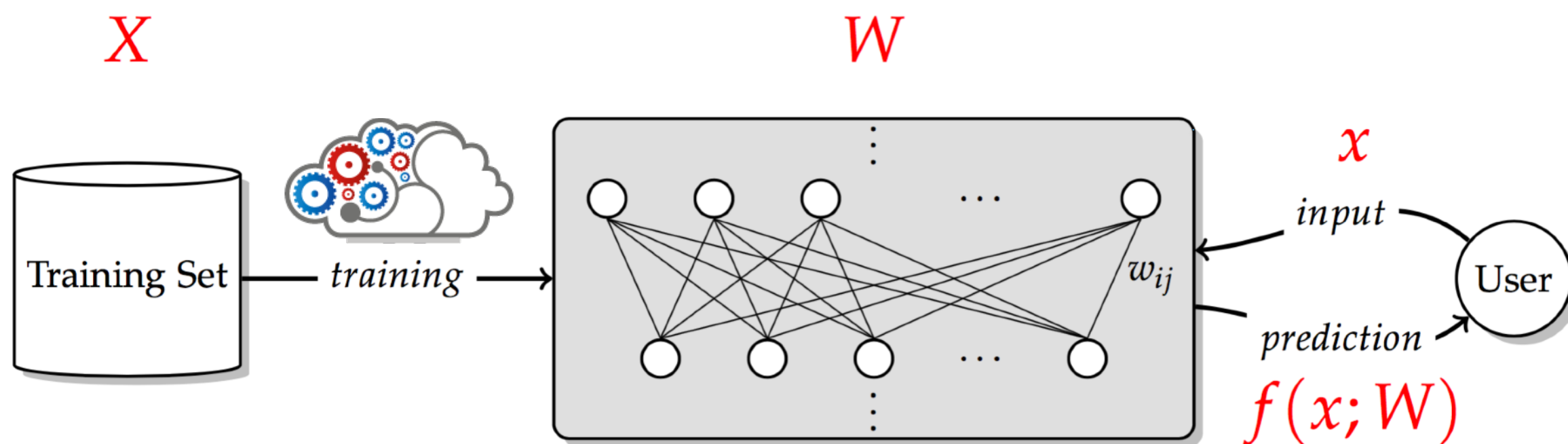
Direct and intentional leakage

- Unauthorized access to data, and data breaches
- Massive data collection

Indirect and unintentional leakage

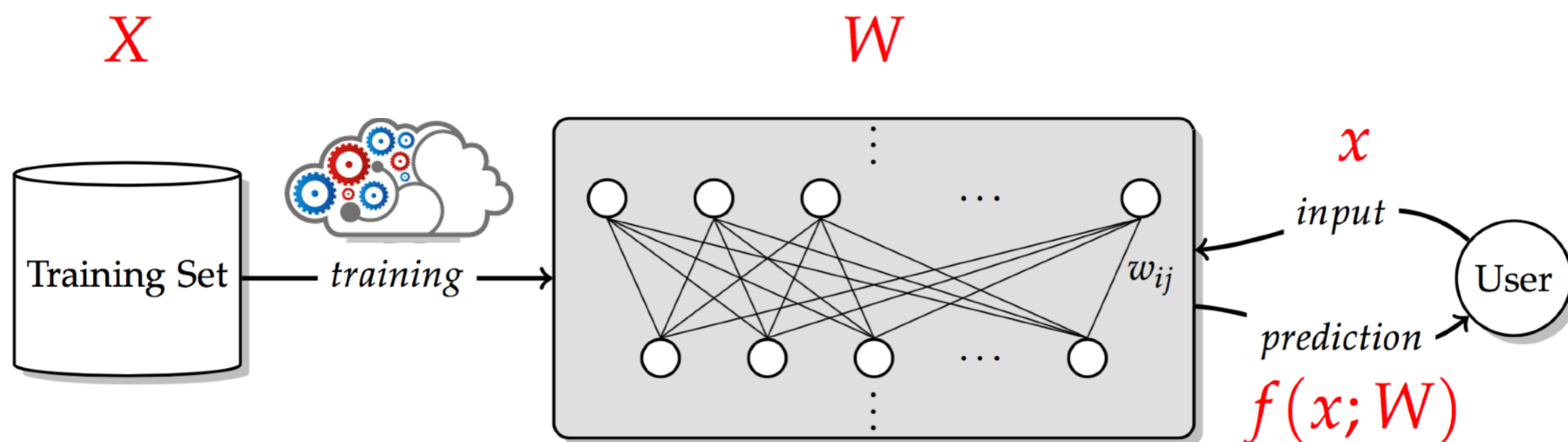
- Meta-data: Data about data
- Data correlated with data
- Computations on data

Privacy Risks in Machine Learning

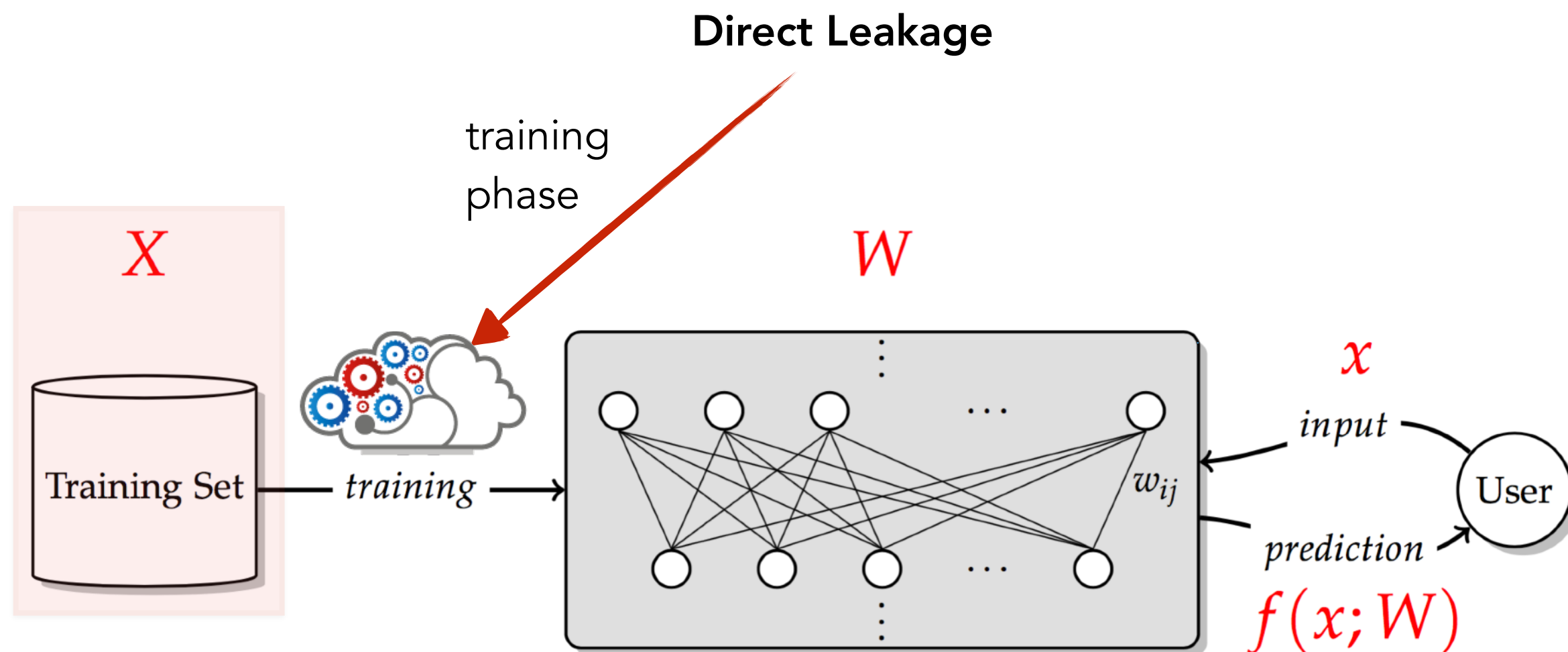


Privacy Risks in Machine Learning

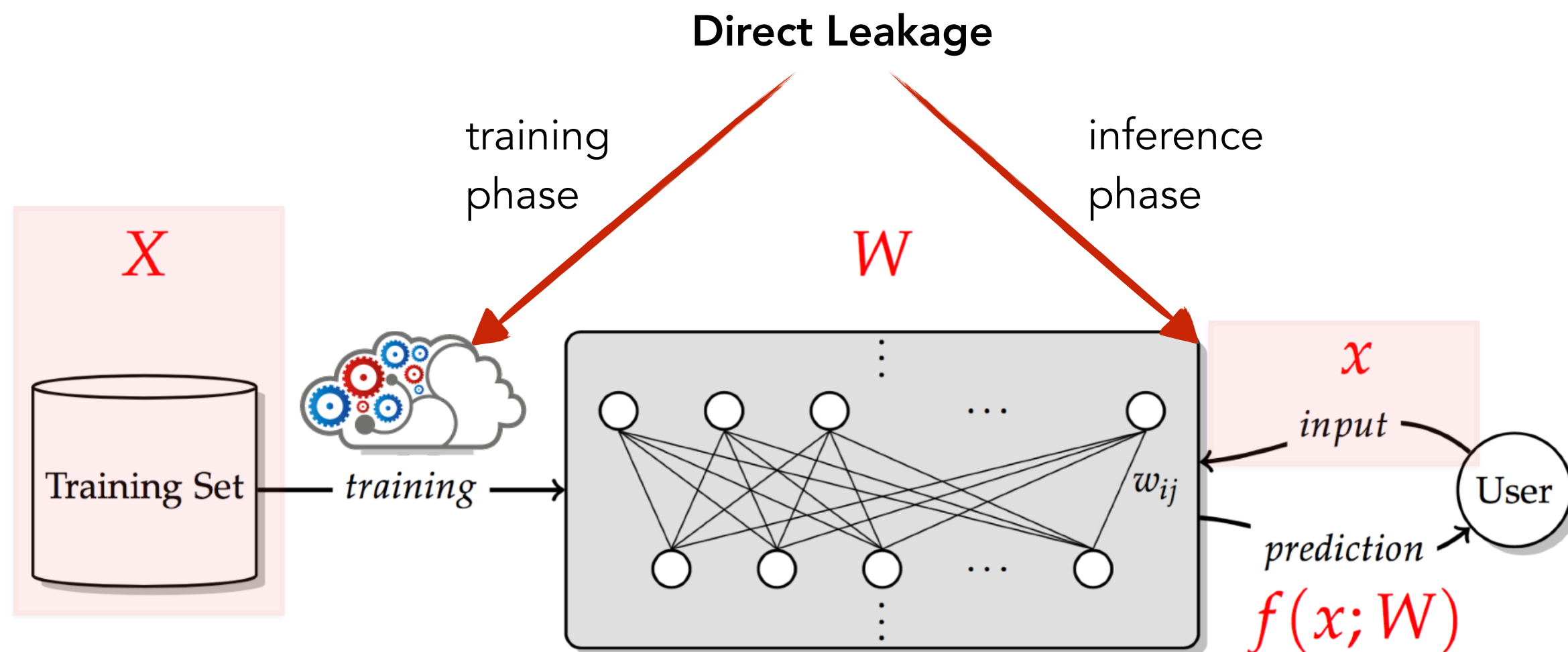
Direct Leakage



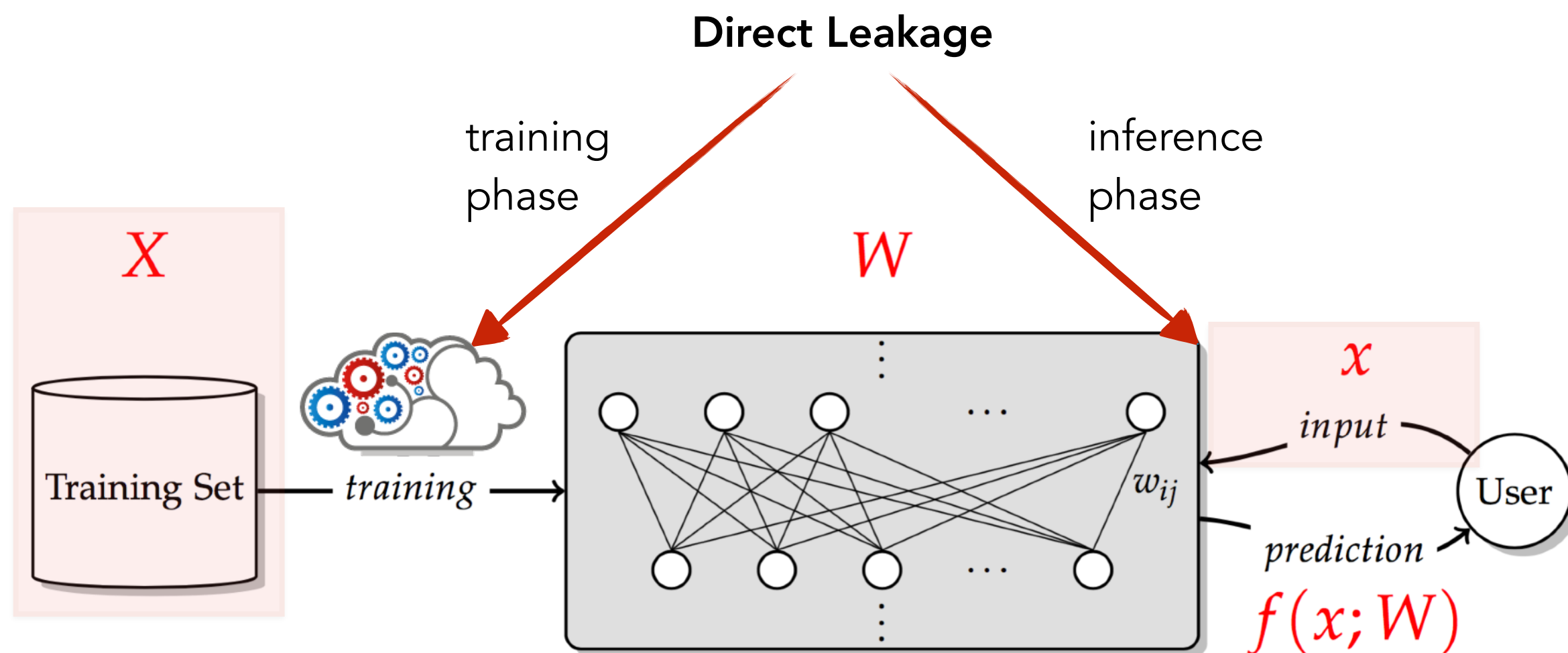
Privacy Risks in Machine Learning



Privacy Risks in Machine Learning

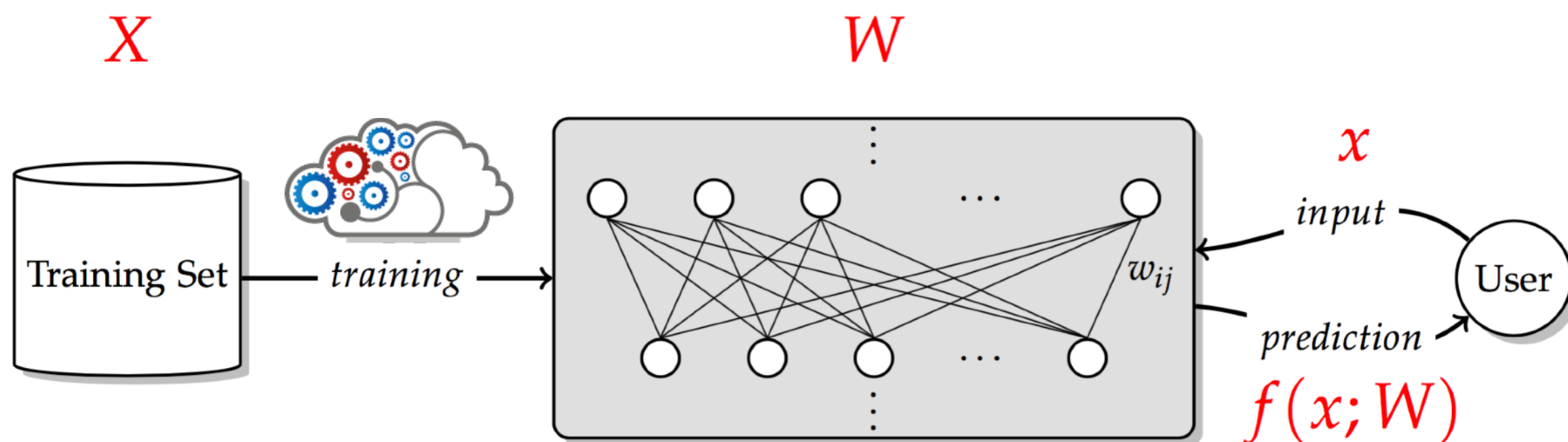


Privacy Risks in Machine Learning



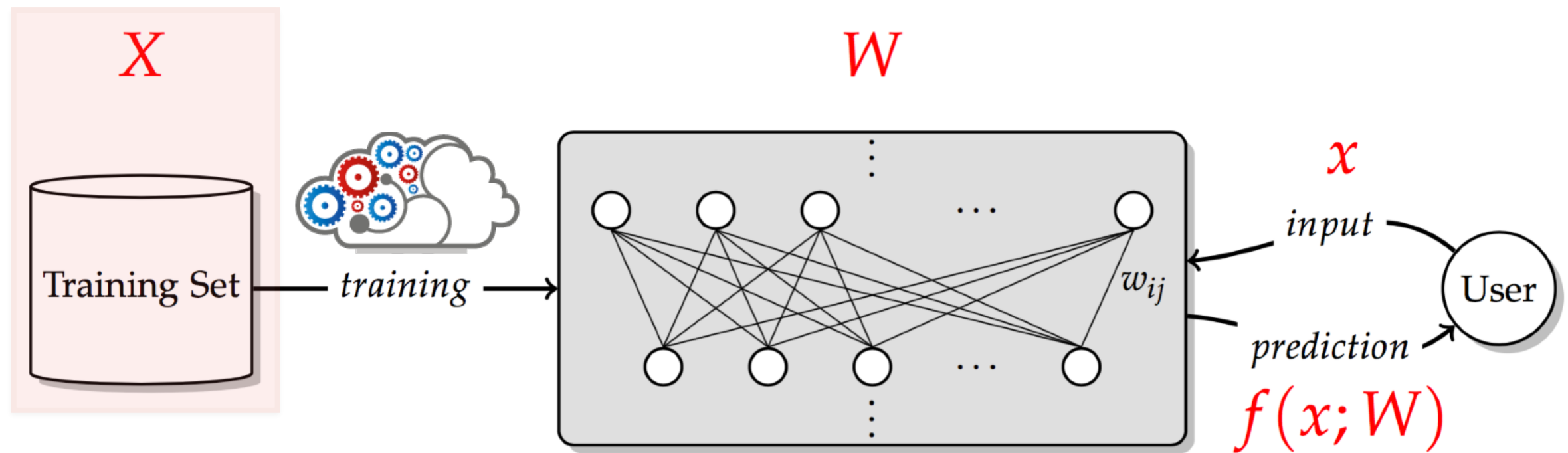
How to prevent leakage? Secure multi-party computation, homomorphic encryption, trusted hardware, ...

Privacy Risks in Machine Learning



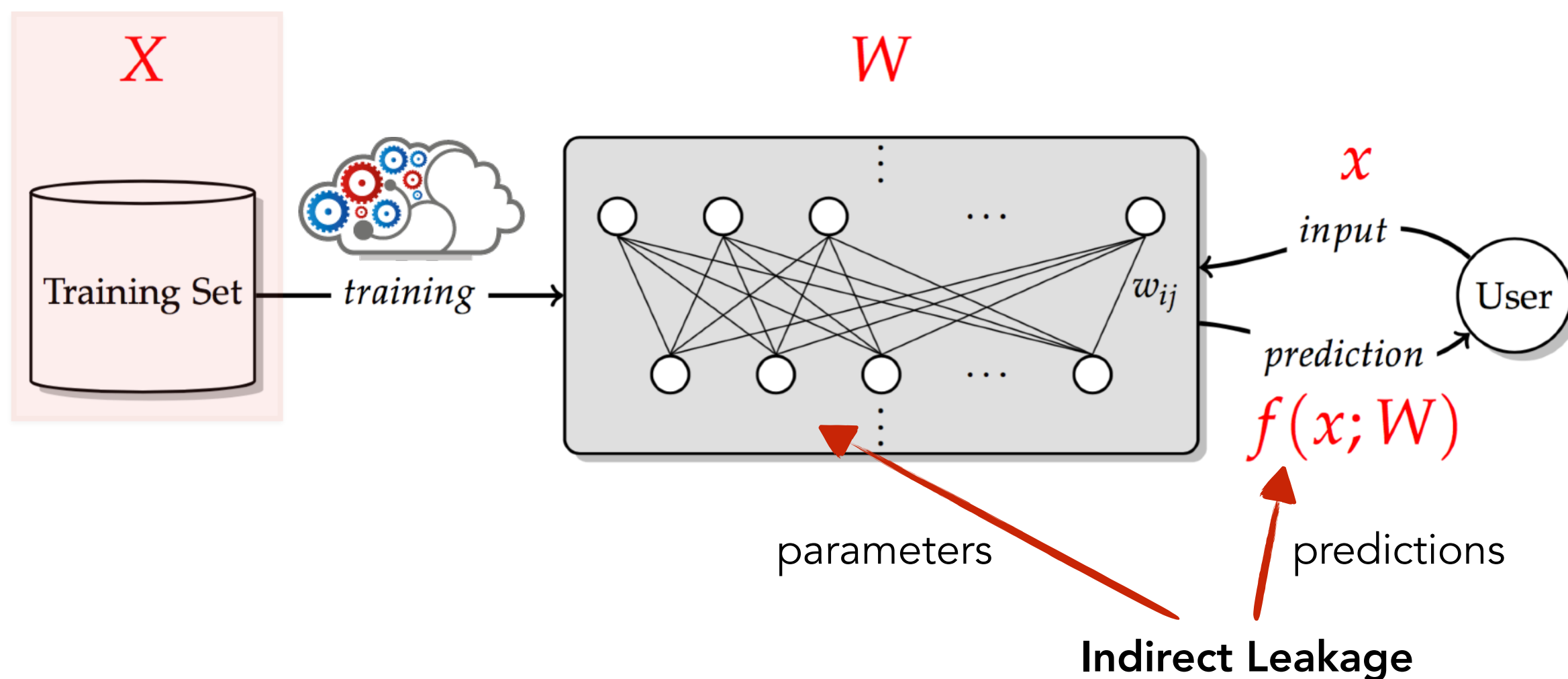
Indirect Leakage

Privacy Risks in Machine Learning



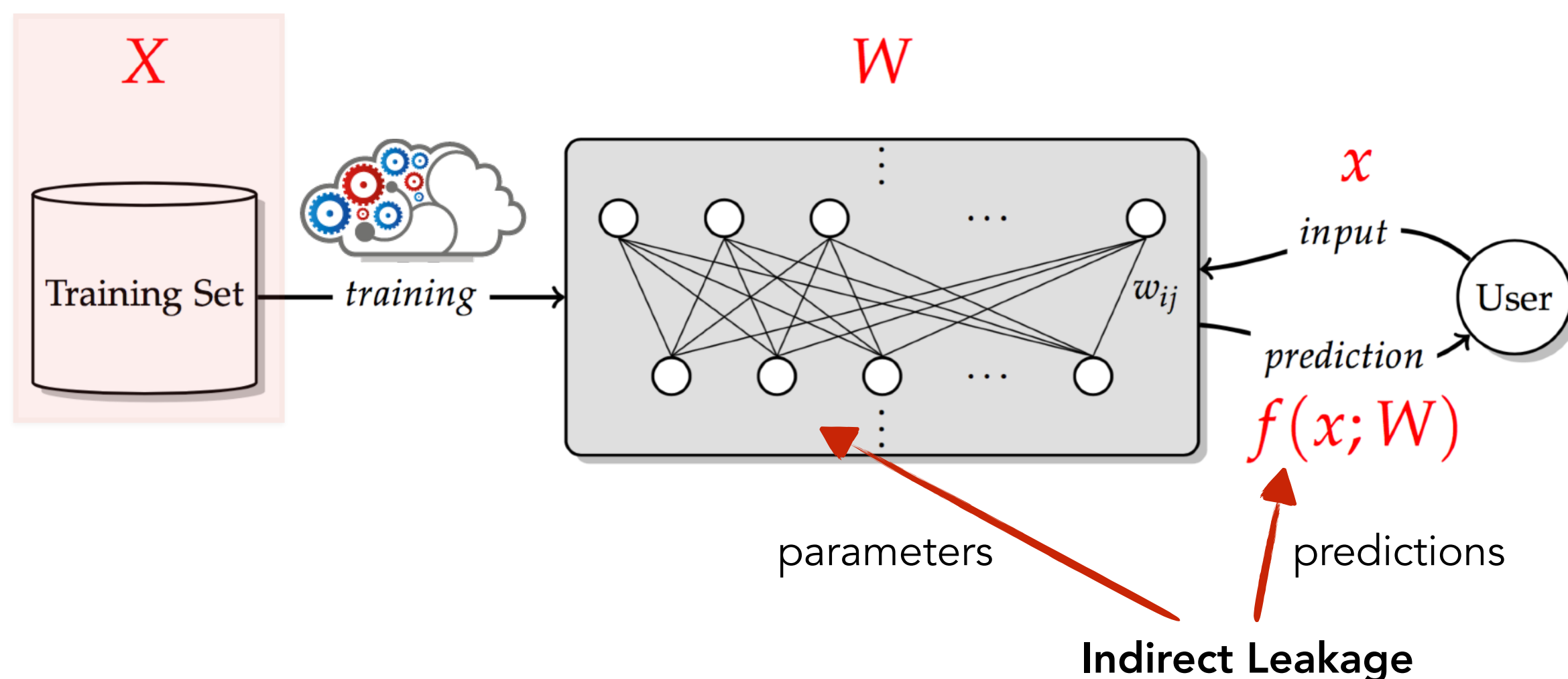
Indirect Leakage

Privacy Risks in Machine Learning



Privacy Risks in Machine Learning

What is leakage? Inferring information about members of X , beyond what can be learned about its underlying distribution

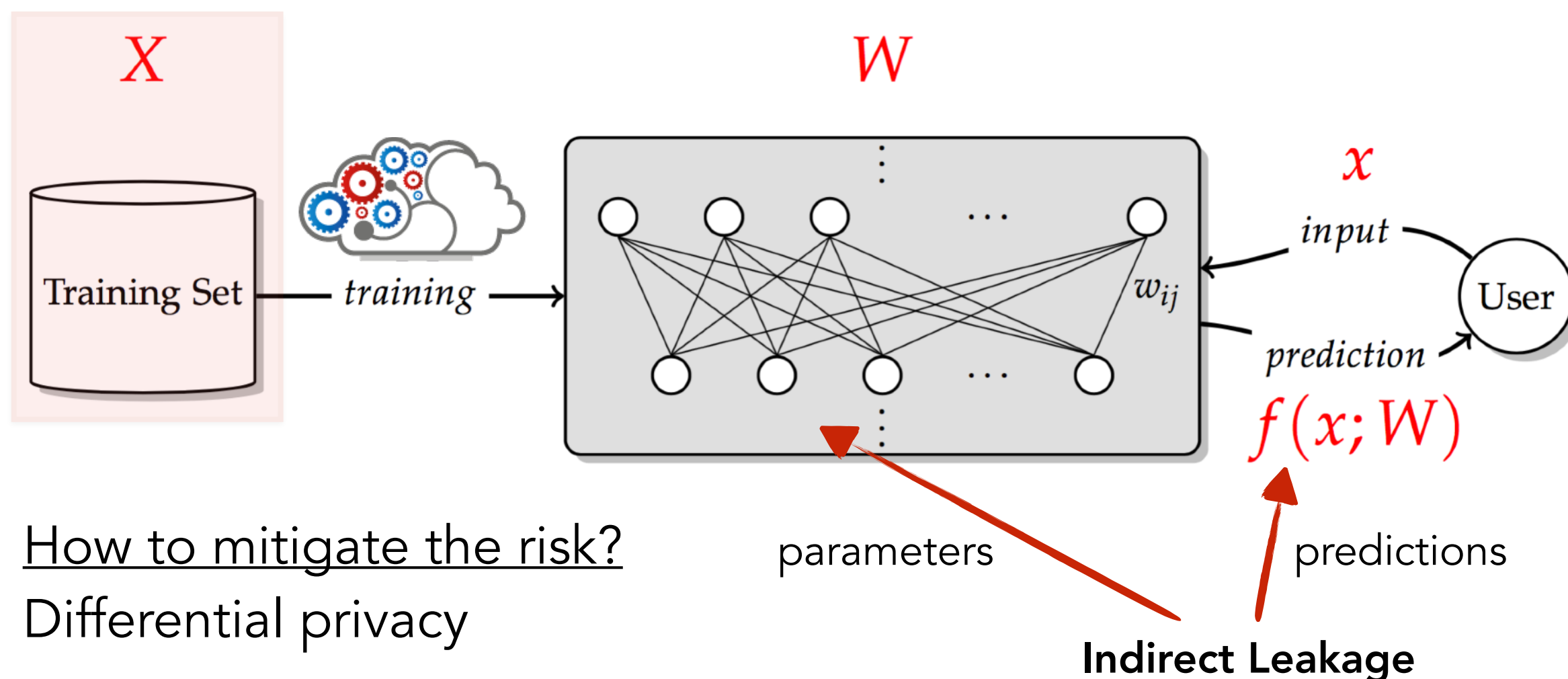


[Shokri, Stronati, Song, Shmatikov] Membership Inference Attacks against Machine Learning Models, SP'17

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

Privacy Risks in Machine Learning

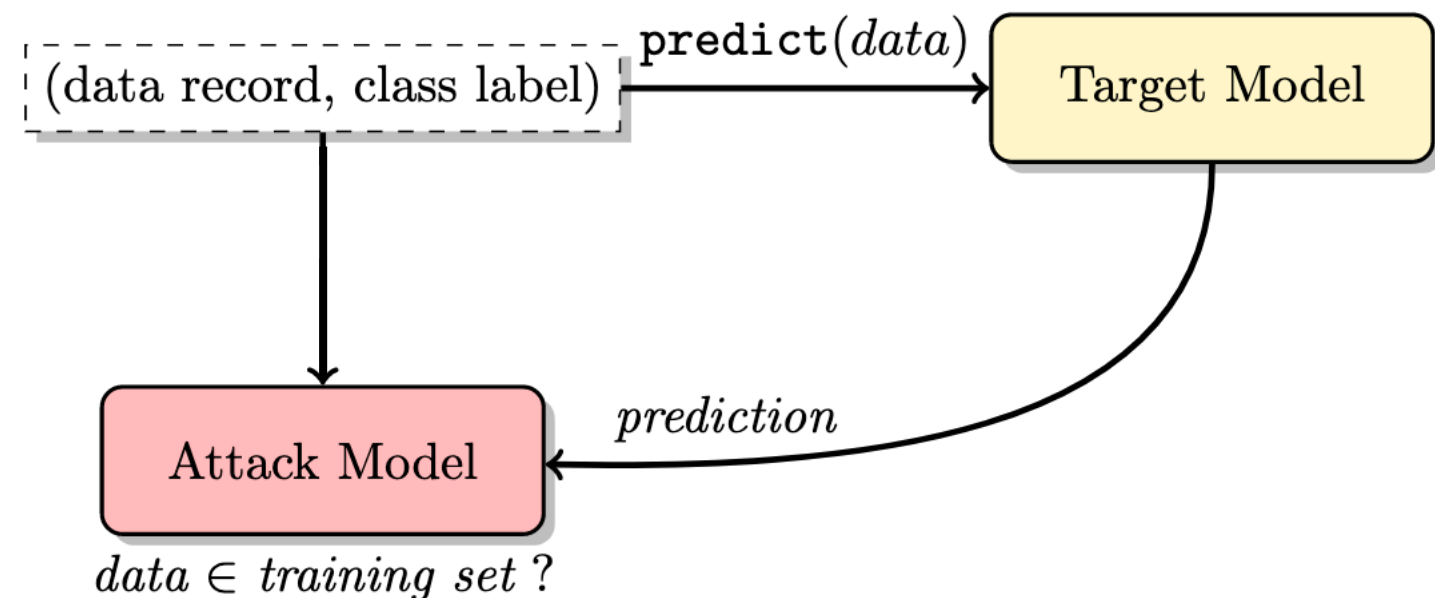
What is leakage? Inferring information about members of X , beyond what can be learned about its underlying distribution



[Shokri, Stronati, Song, Shmatikov] Membership Inference Attacks against Machine Learning Models, SP'17
[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

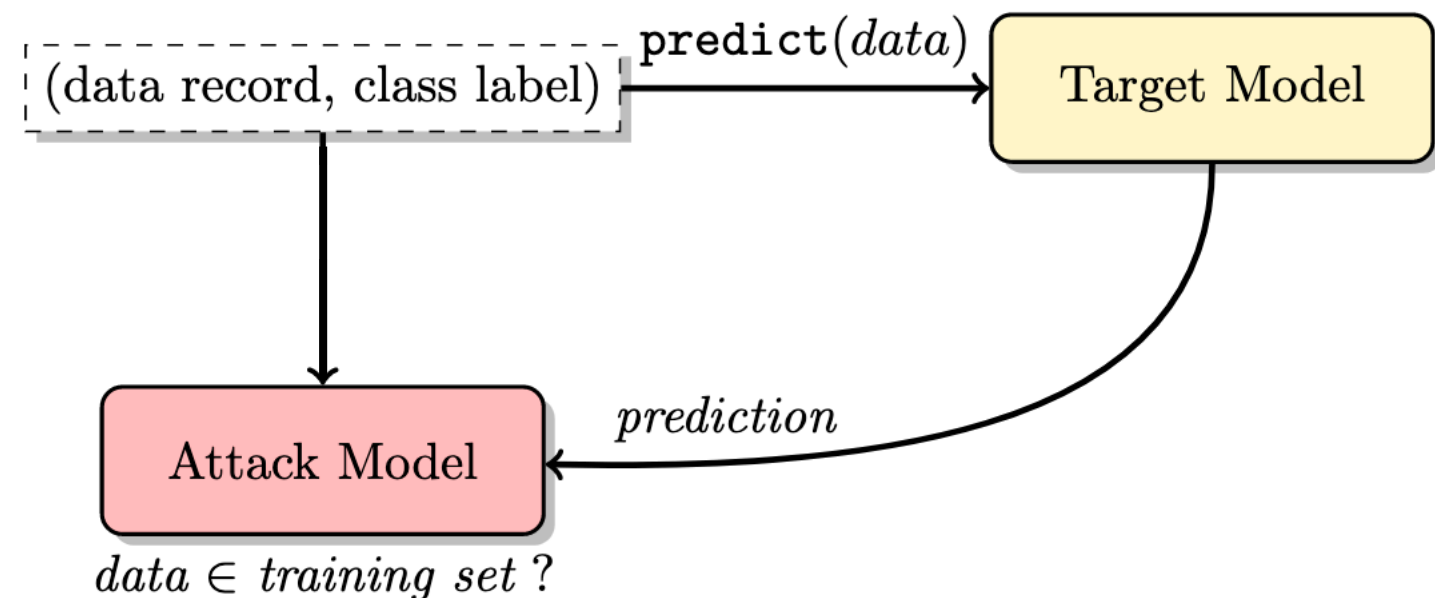
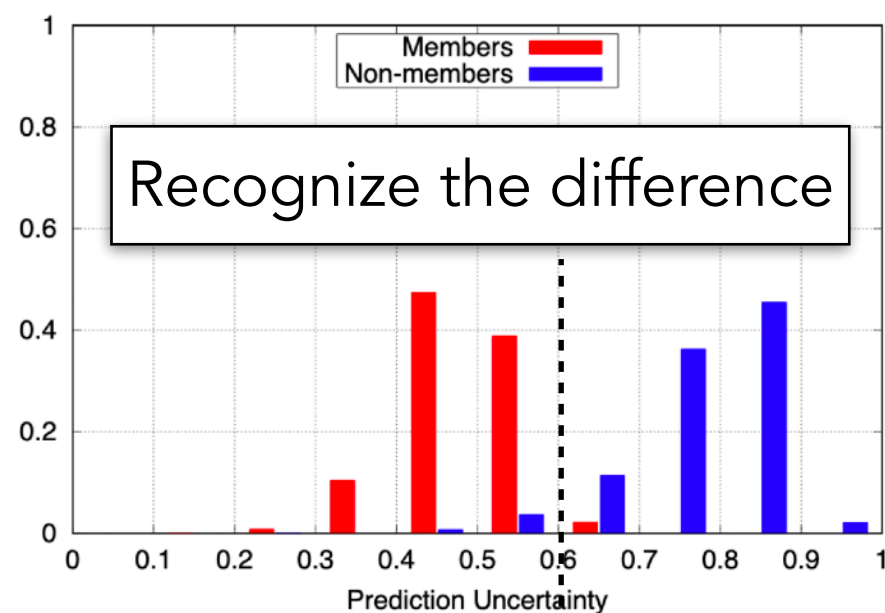
How to Quantify the Leakage?

- Indistinguishability game: Can an adversary distinguish between two models that are trained on two neighboring datasets (one includes an extra data point x)?
- Membership inference: Given a model, can an adversary infer whether data point x is part of its training set?



How to Quantify the Leakage?

- Indistinguishability game: Can an adversary distinguish between two models that are trained on two neighboring datasets (one includes an extra data point x)?
- Membership inference: Given a model, can an adversary infer whether data point x is part of its training set?

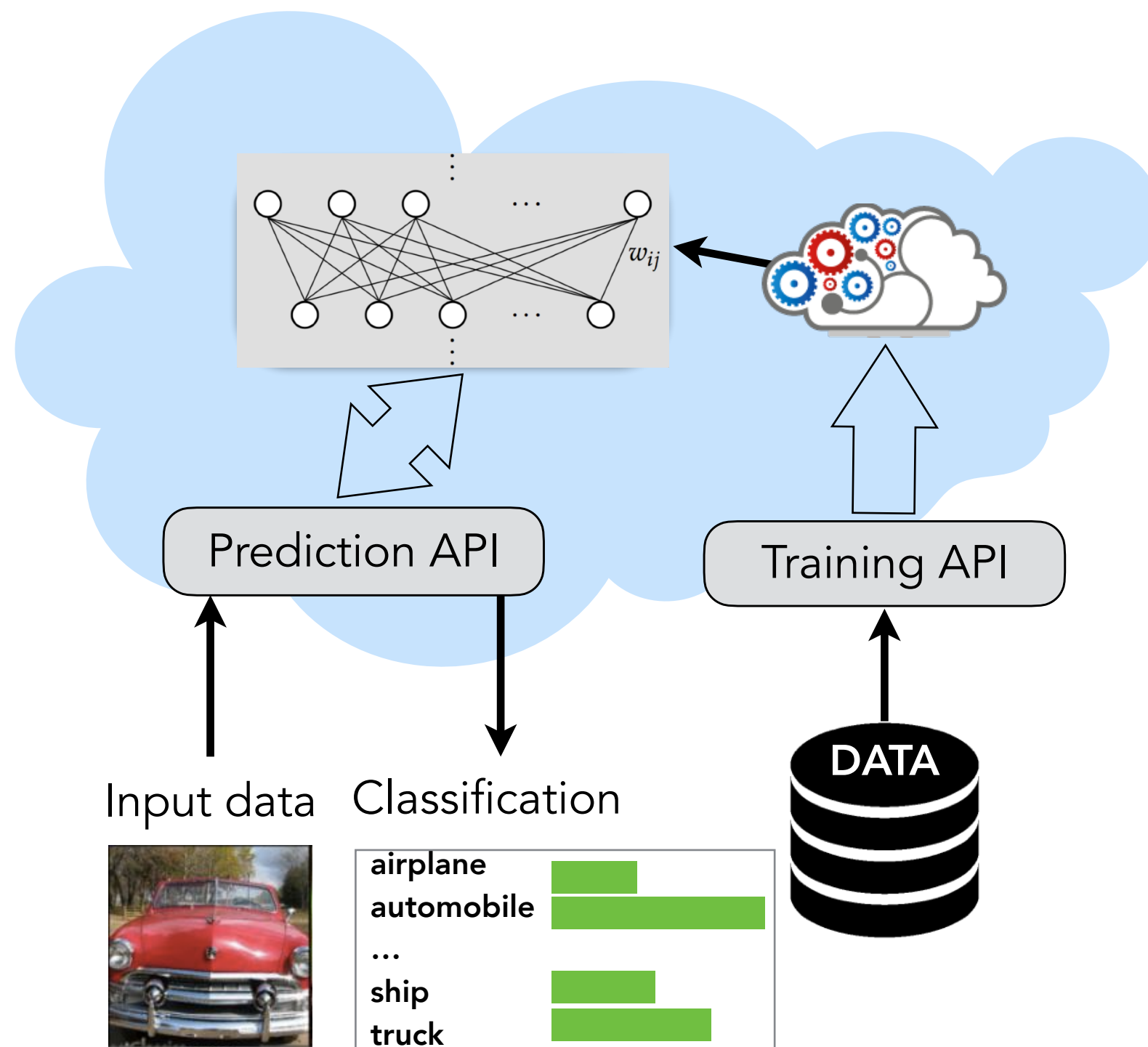


Membership Inference Attacks against Classification Models

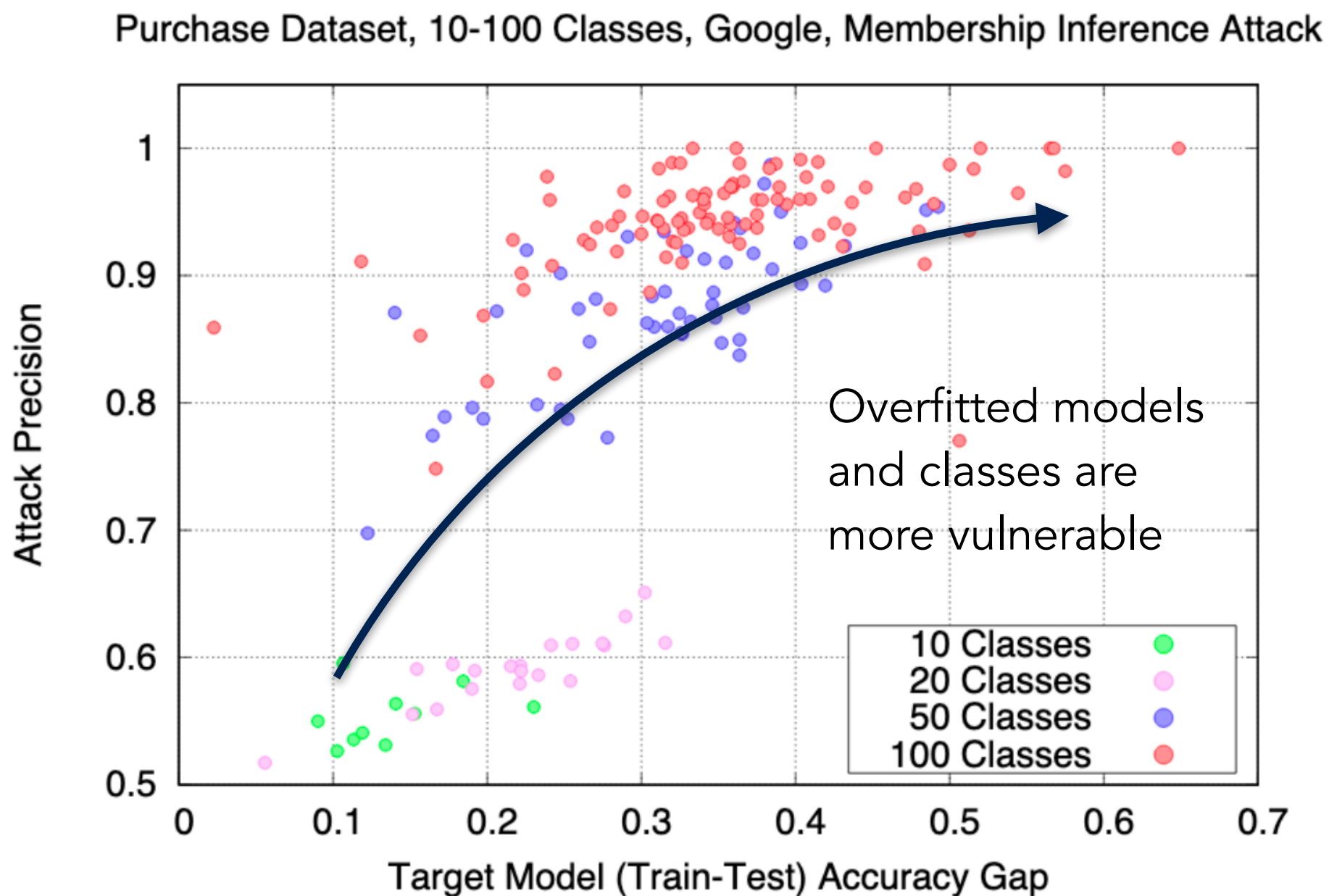
Machine Learning
as a Service



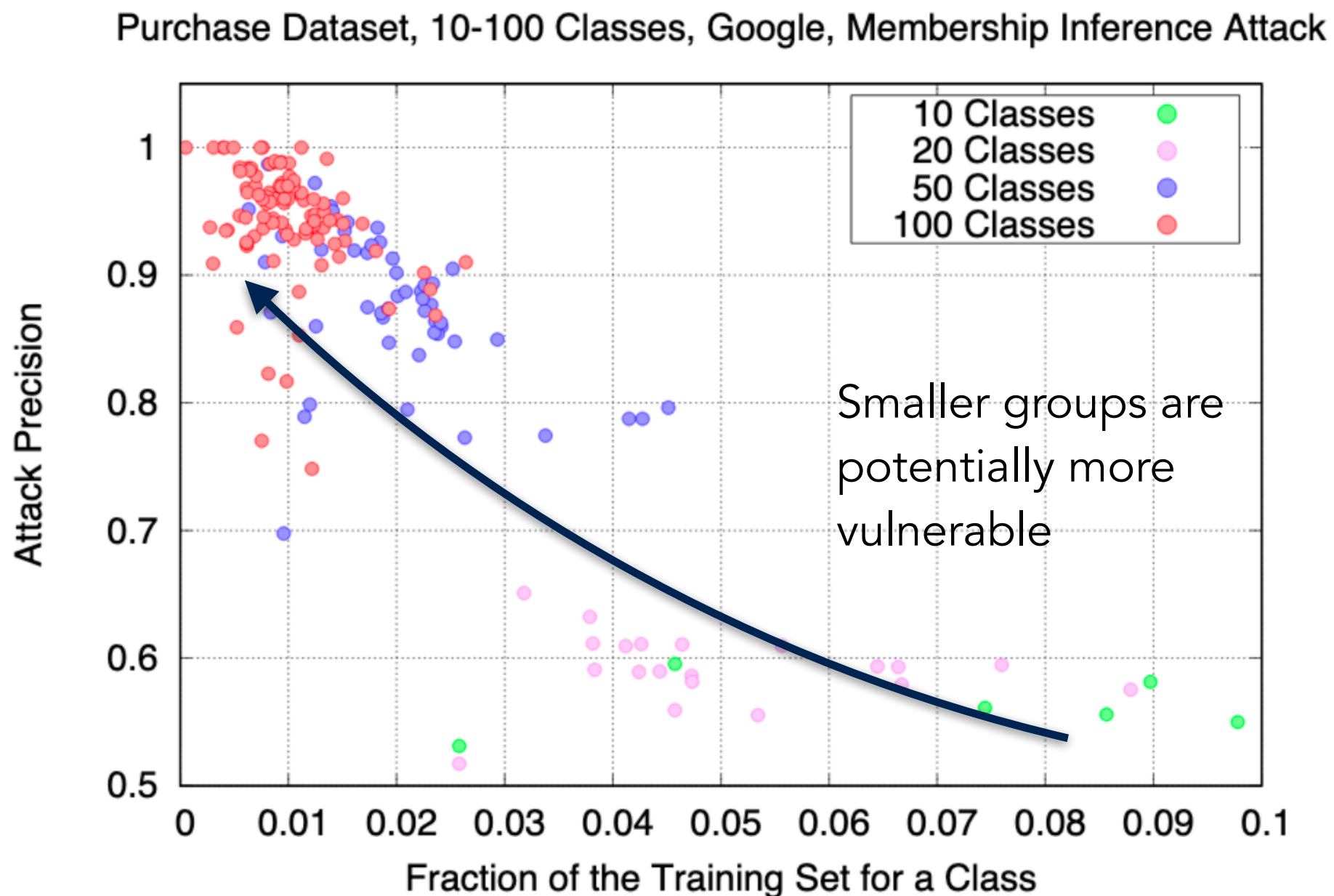
**Membership
Inference Attack**
Accuracy:
~ 90%



Privacy Leakage due to Overfitting



Disparate Privacy Vulnerability



White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%			
CIFAR100	ResNet	89%	73%			
CIFAR100	DenseNet	100%	82%			

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%			
CIFAR100	ResNet	89%	73%			
CIFAR100	DenseNet	100%	82%			

High generalizability
to test data

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%	74.2%	74.6%	75.1%
CIFAR100	ResNet	89%	73%	62.2%	62.2%	64.3%
CIFAR100	DenseNet	100%	82%	67.7%	67.7%	74.3%

High generalizability
to test data

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%	74.2%	74.6%	75.1%
CIFAR100	ResNet	89%	73%	62.2%	62.2%	64.3%
CIFAR100	DenseNet	100%	82%	67.7%	67.7%	74.3%

High generalizability
to test data

Low privacy
(Significant leakage
through parameters)

White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%	74.2%	74.6%	75.1%
CIFAR100	ResNet	89%	73%	62.2%	62.2%	64.3%
CIFAR100	DenseNet	100%	82%	67.7%	67.7%	74.3%

Large capacity (points to DenseNet architecture)
High generalizability to test data (points to DenseNet test accuracy)
Low privacy (Significant leakage through parameters) (points to DenseNet white-box (Gradients) accuracy)

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

White-box Privacy Analysis

- Leakage through parameters (white-box) vs. predictions (black-box)

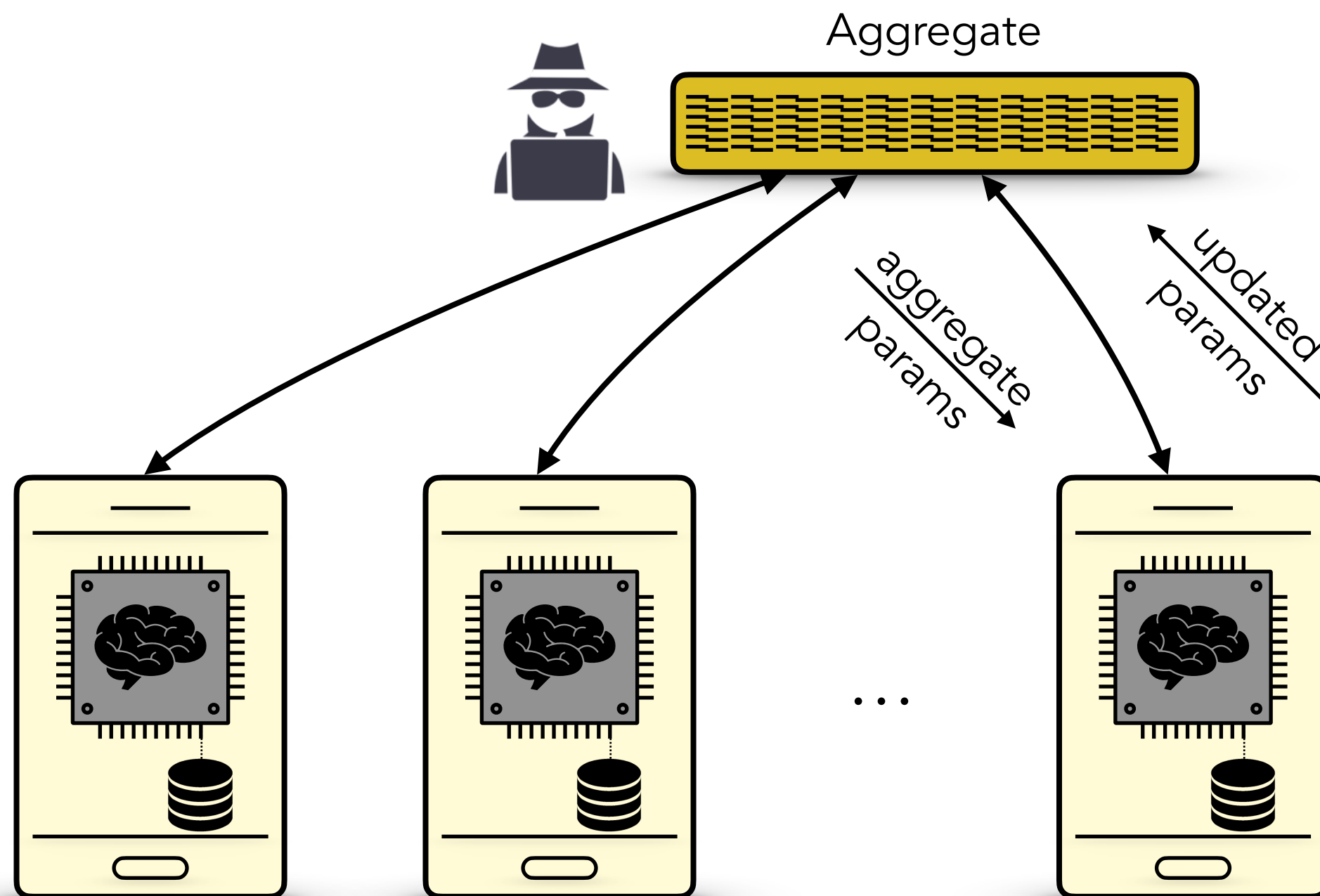
Most accurate pre-trained models				Mem inference attack accuracy		
Dataset	Architecture	Train Accuracy	Test Accuracy	Black-box	White-box (Outputs)	White-box (Gradients)
CIFAR100	Alexnet	99%	44%	74.2%	74.6%	75.1%
CIFAR100	ResNet	89%	73%	62.2%	62.2%	64.3%
CIFAR100	DenseNet	100%	82%	67.7%	67.7%	74.3%

Large capacity
High generalizability to test data
Low privacy (Significant leakage through parameters)

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

[Feldman] Does Learning Require **Memorization**? A Short Tale about a Long Tail, STOC'20

Decentralized (Federated) Learning



[Shokri and Shmatikov] Privacy-Preserving Deep Learning, CCS'15

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

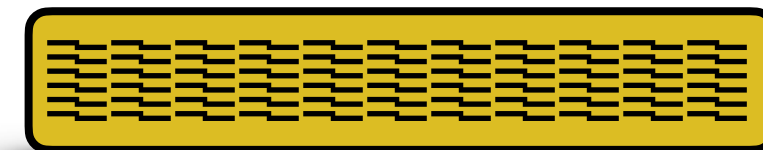
[Melis, Song, De Cristofaro, Shmatikov] Exploiting Unintended Feature Leakage in Collaborative Learning, SP'19

Decentralized (Federated) Learning

Adversary can observe multiple snapshots of the model

Observed Epochs	Attack Accuracy
5, 10, 15, 20, 25	57.4%
10, 20, 30, 40, 50	76.5%
50, 100, 150, 200, 250	79.5%
100, 150, 200, 250, 300	85.1%

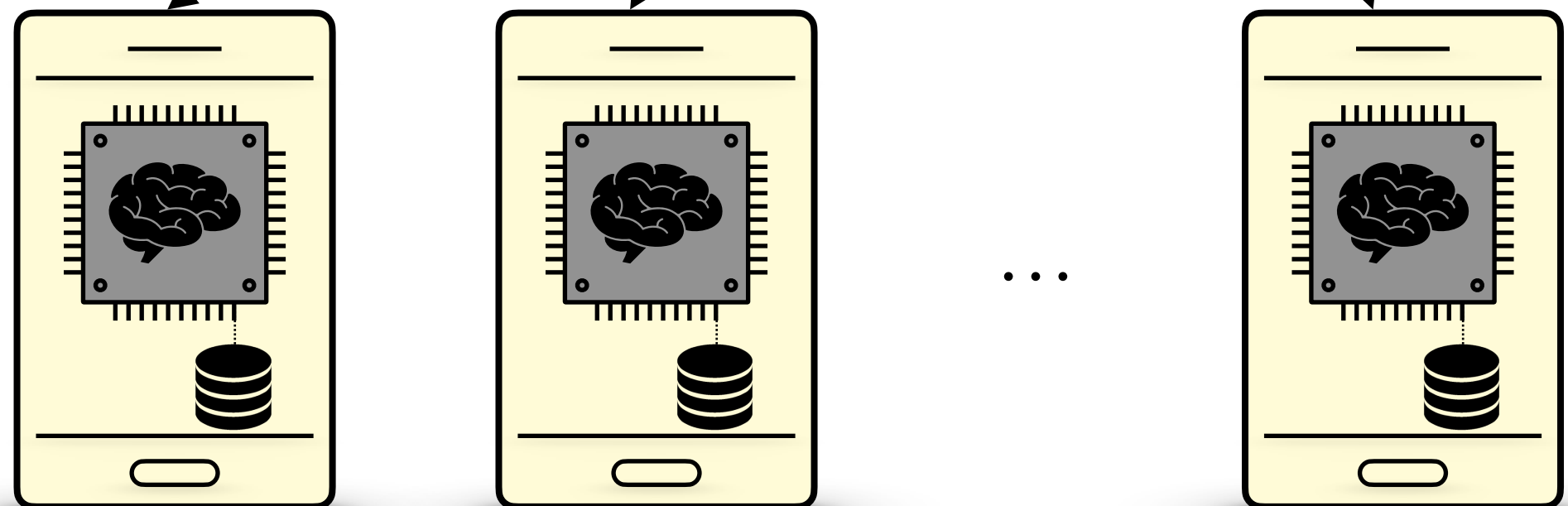
Aggregate



aggregate
params

updated
params

CIFAR100-Alexnet



[Shokri and Shmatikov] Privacy-Preserving Deep Learning, CCS'15

[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

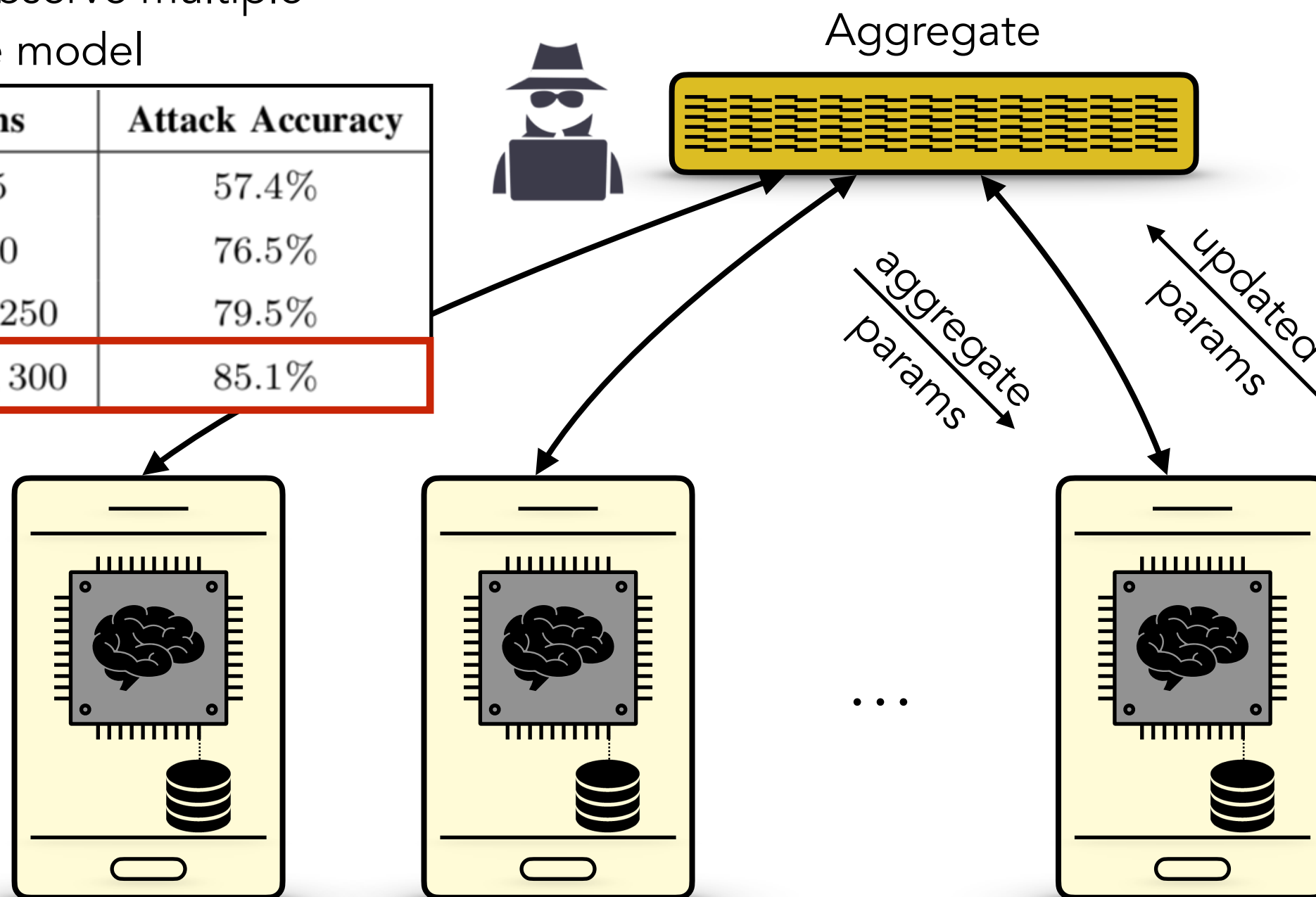
[Melis, Song, De Cristofaro, Shmatikov] Exploiting Unintended Feature Leakage in Collaborative Learning, SP'19

Decentralized (Federated) Learning

Adversary can observe multiple snapshots of the model

Observed Epochs	Attack Accuracy
5, 10, 15, 20, 25	57.4%
10, 20, 30, 40, 50	76.5%
50, 100, 150, 200, 250	79.5%
100, 150, 200, 250, 300	85.1%

CIFAR100-Alexnet

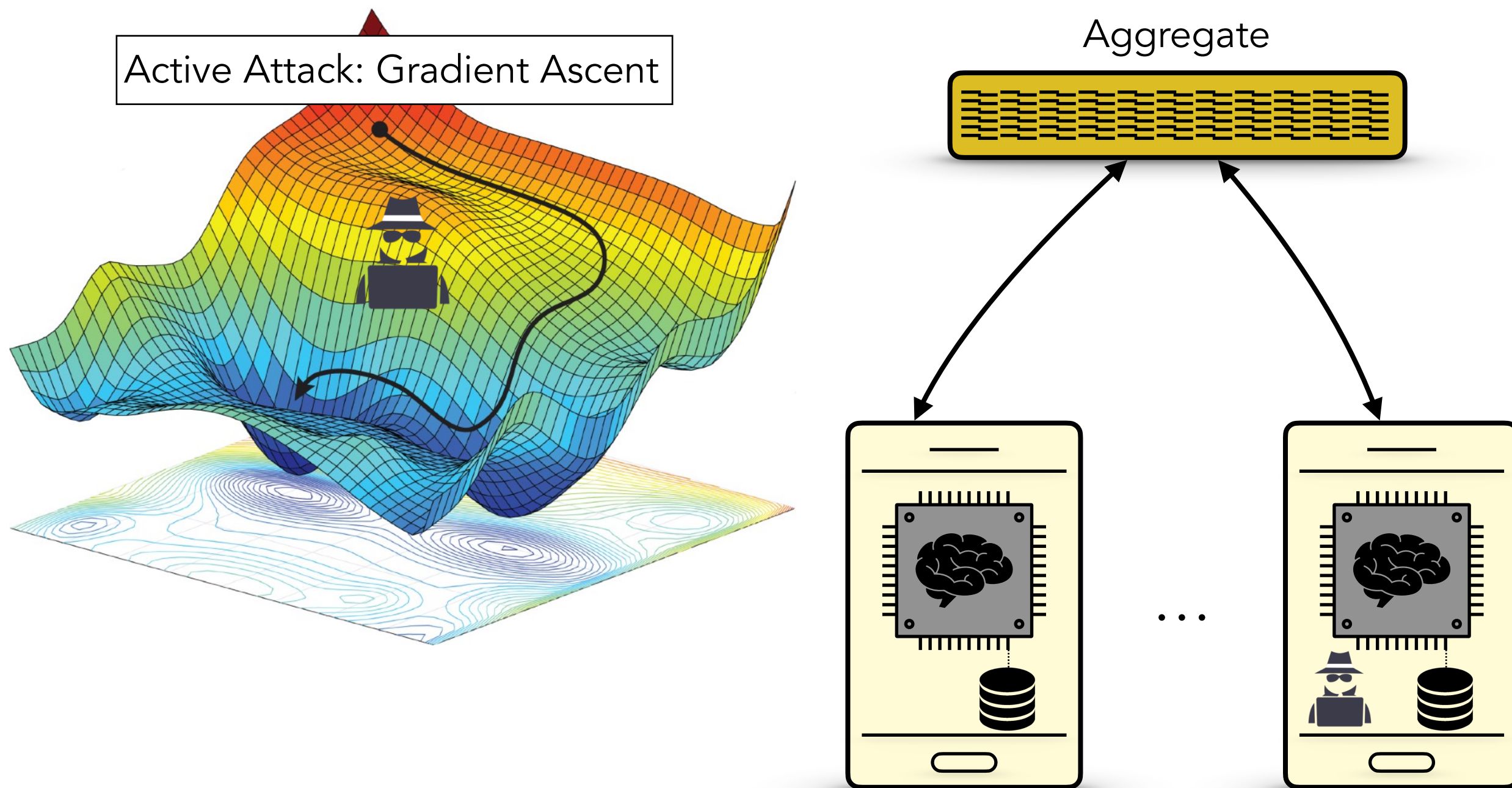


[Shokri and Shmatikov] Privacy-Preserving Deep Learning, CCS'15

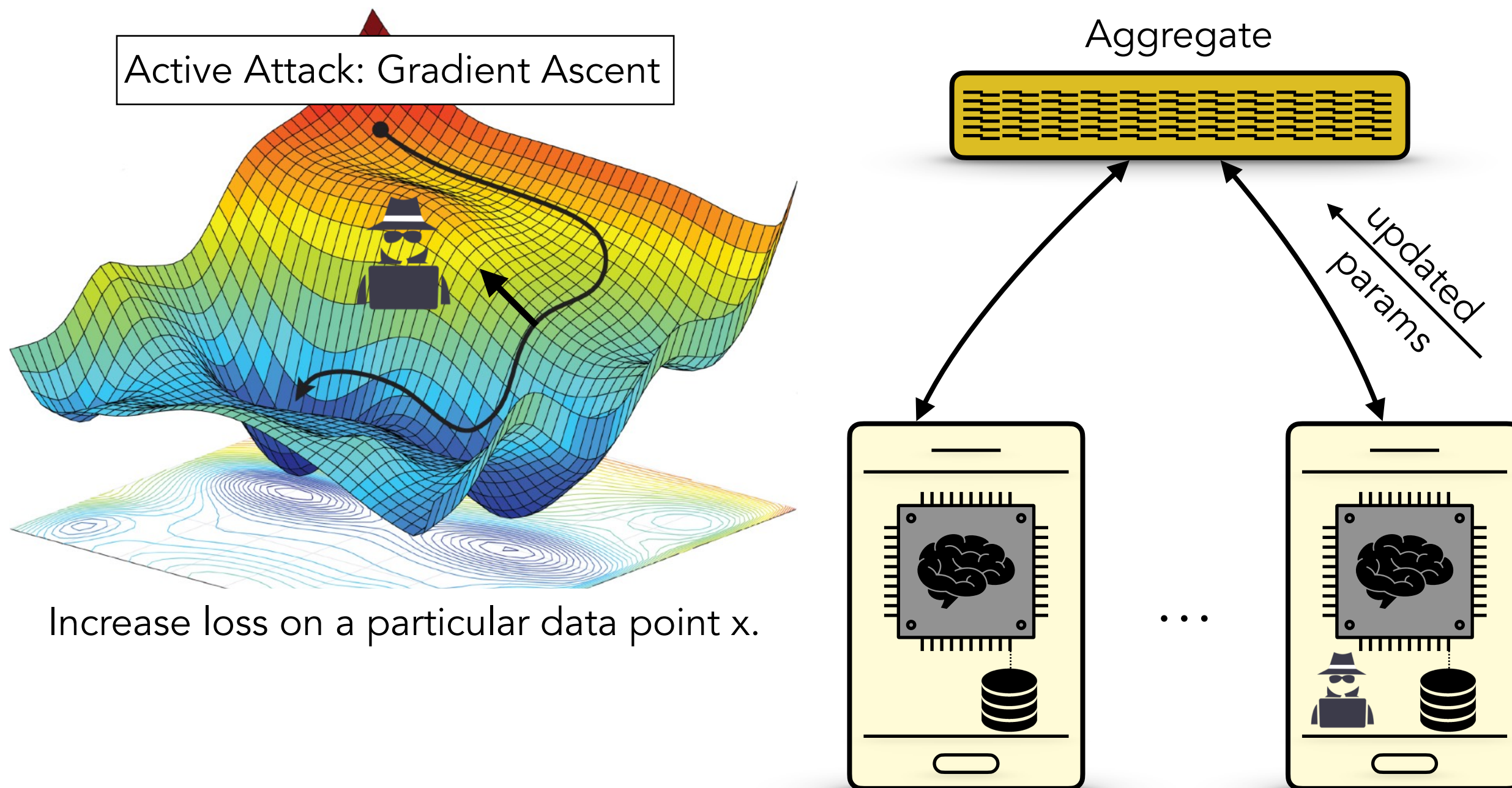
[Nasr, Shokri, Houmansadr] Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning, SP'19

[Melis, Song, De Cristofaro, Shmatikov] Exploiting Unintended Feature Leakage in Collaborative Learning, SP'19

Decentralized (Federated) Learning

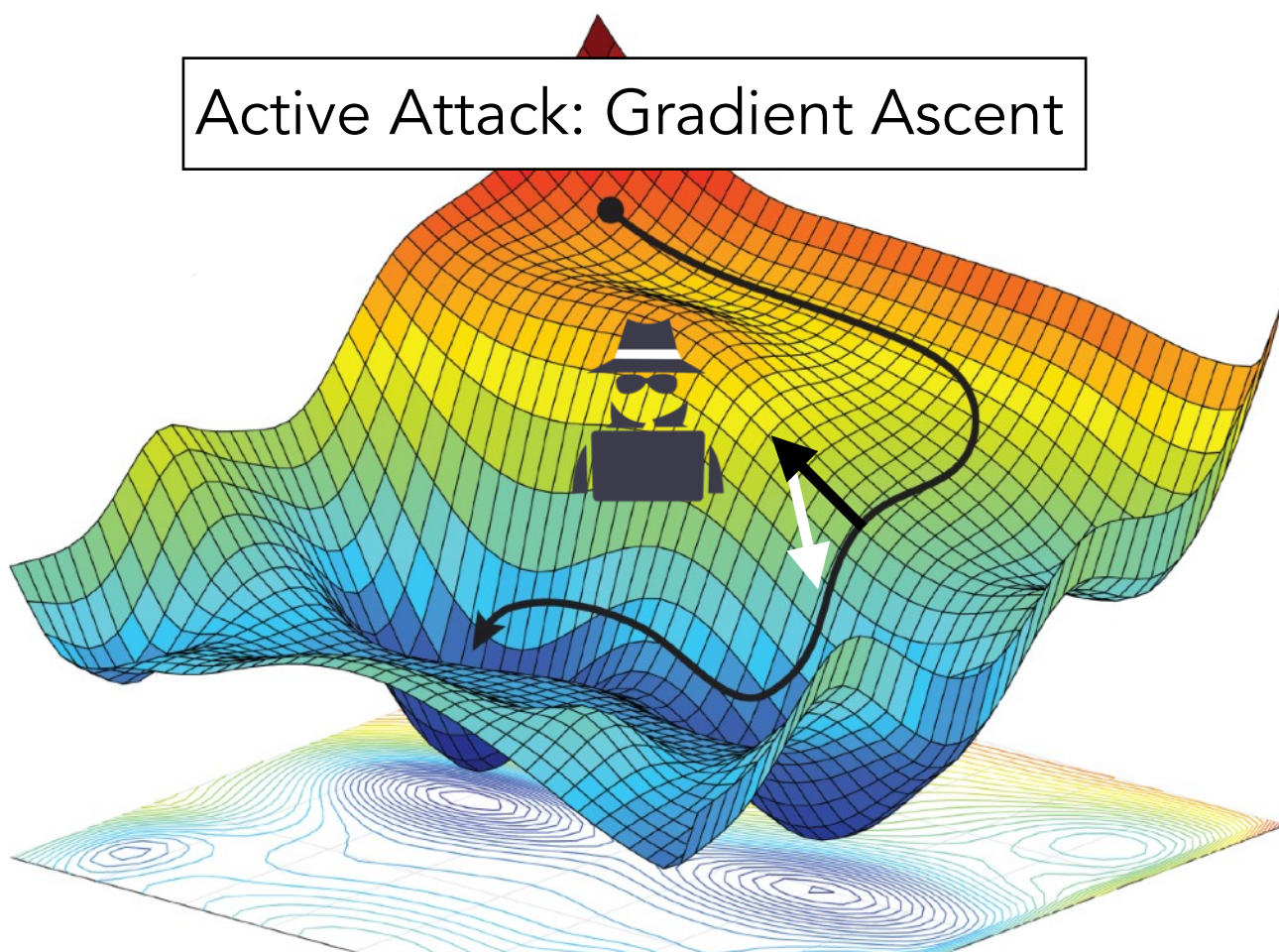


Decentralized (Federated) Learning



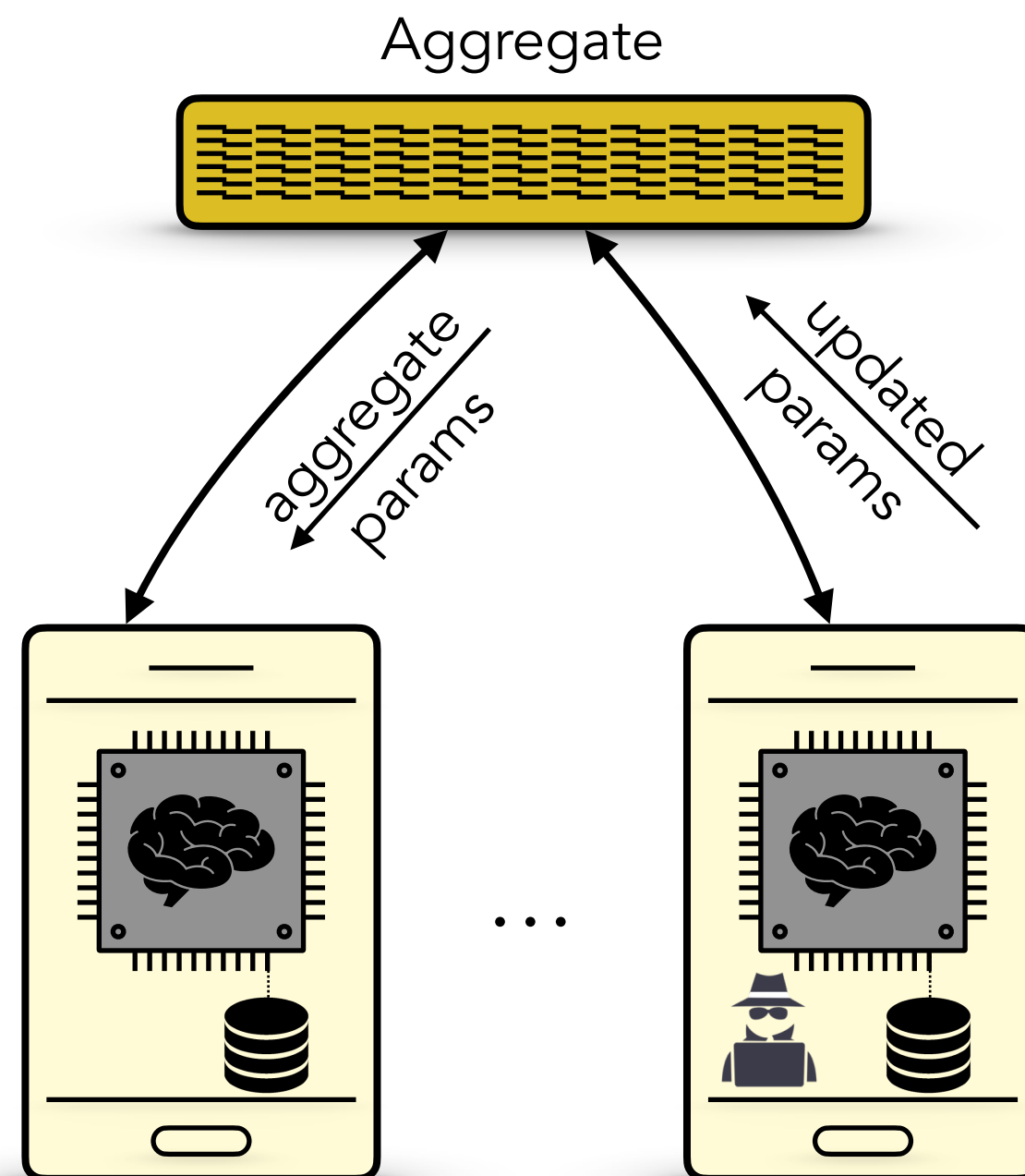
Decentralized (Federated) Learning

Active Attack: Gradient Ascent





Increase loss on a particular data point x .

A participant correct it back (by running gradient descent locally) only if x is part of its training set. => **membership leakage**



AI Regulations - Data Protection

- “... membership inferences show that AI models can inadvertently contain personal data”  Information Commissioner's Office
- “Attacks that reveal confidential information about the data include membership inference whereby ...”  National Institute of Standards and Technology
U.S. Department of Commerce
- “..... ensuring that privacy and personal data are adequately protected during the use of AI”
- “..... ensuring that AI systems are resilient to overt attacks and subtle attacks that manipulate data or algorithms....”
- “...should consider the risks to data throughout the design, development, and operation of an AI system”

On Artificial Intelligence - A European Approach to excellence and trust - Feb 2020

The White House Memo on Guidance for Regulation of Artificial Intelligence Applications - Jan 2020

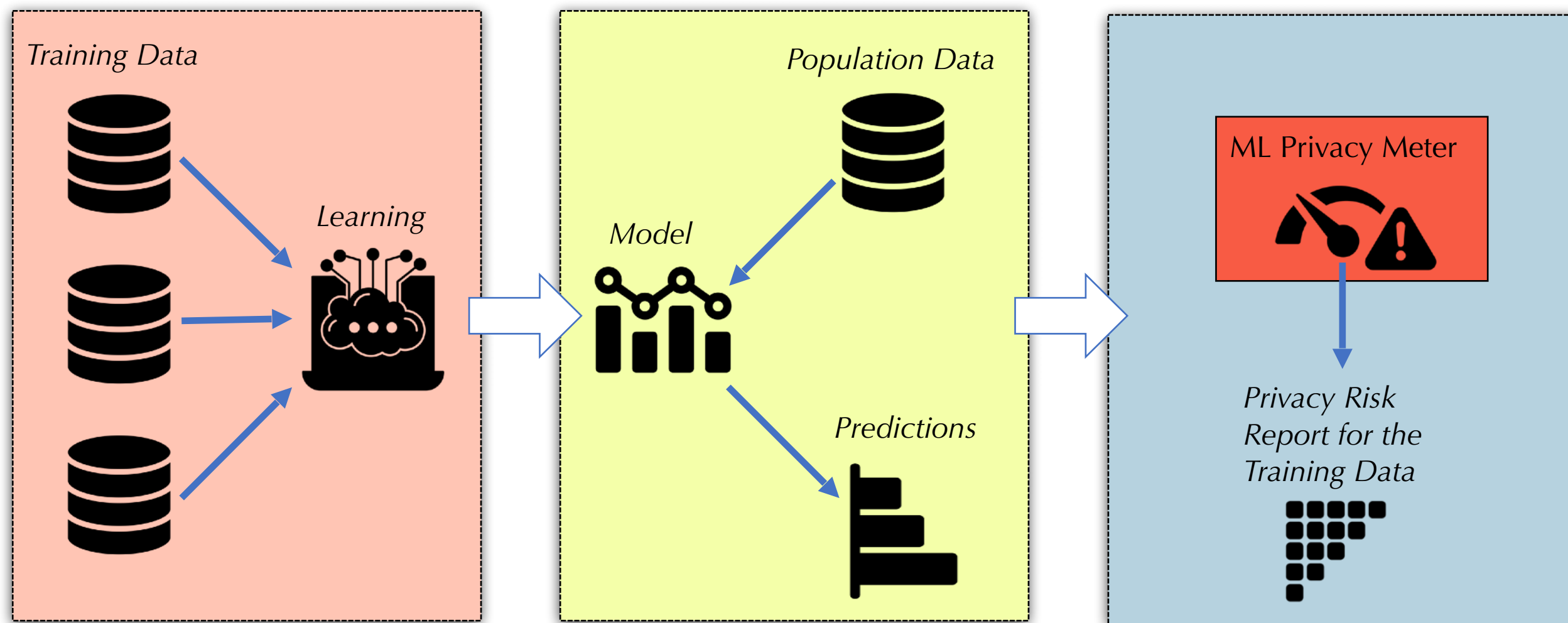
Guidance on the AI auditing framework Draft guidance for consultation. Information Commissioner's Office

A Taxonomy and Terminology of Adversarial Machine Learning. Draft NISTIR 8269

Data Protection Impact Assessment



Tool: ML Privacy Meter

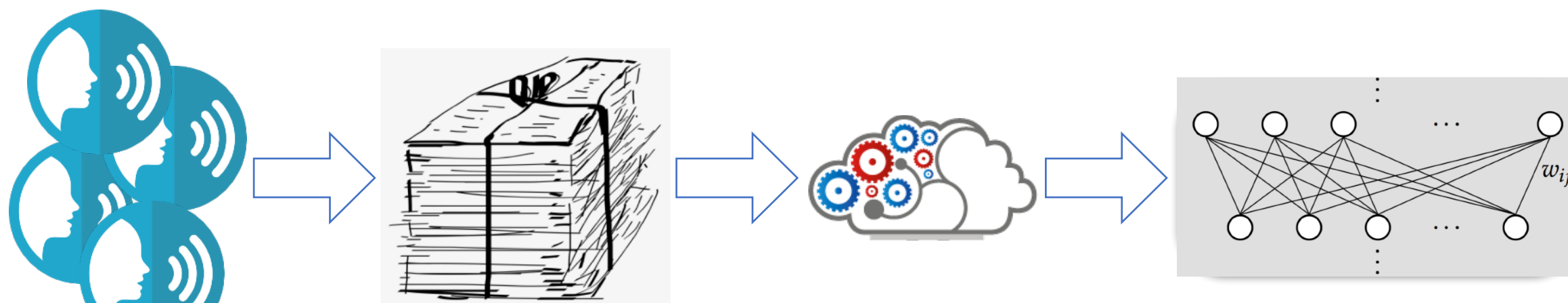


ML Privacy Meter is a Python library (`ml_privacy_meter`) that enables quantifying the privacy risks of machine learning models. https://github.com/privacytrustlab/ml_privacy_meter

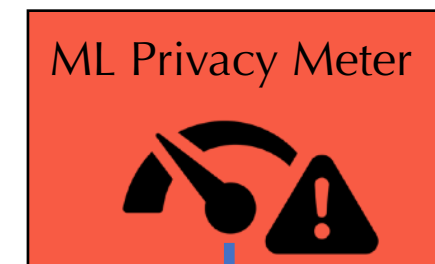


ML Privacy Meter

Example: NLP Models



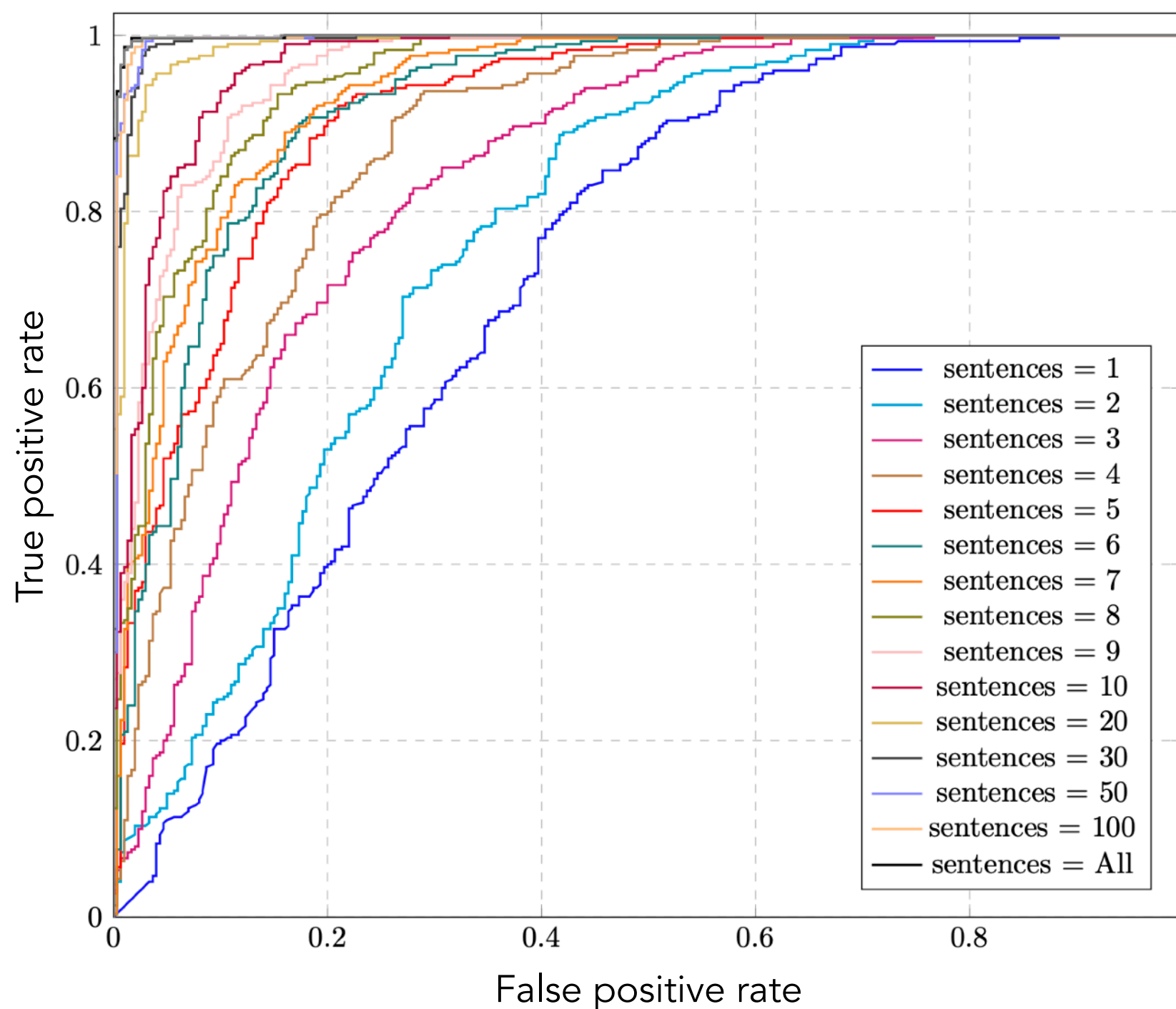
- How much does the model leak about the sentences of a particular author/speaker? What about the membership of the author in the training set (based on known samples)?
- Which samples are leaked?



*Privacy Risk
Report for the
Training Data*

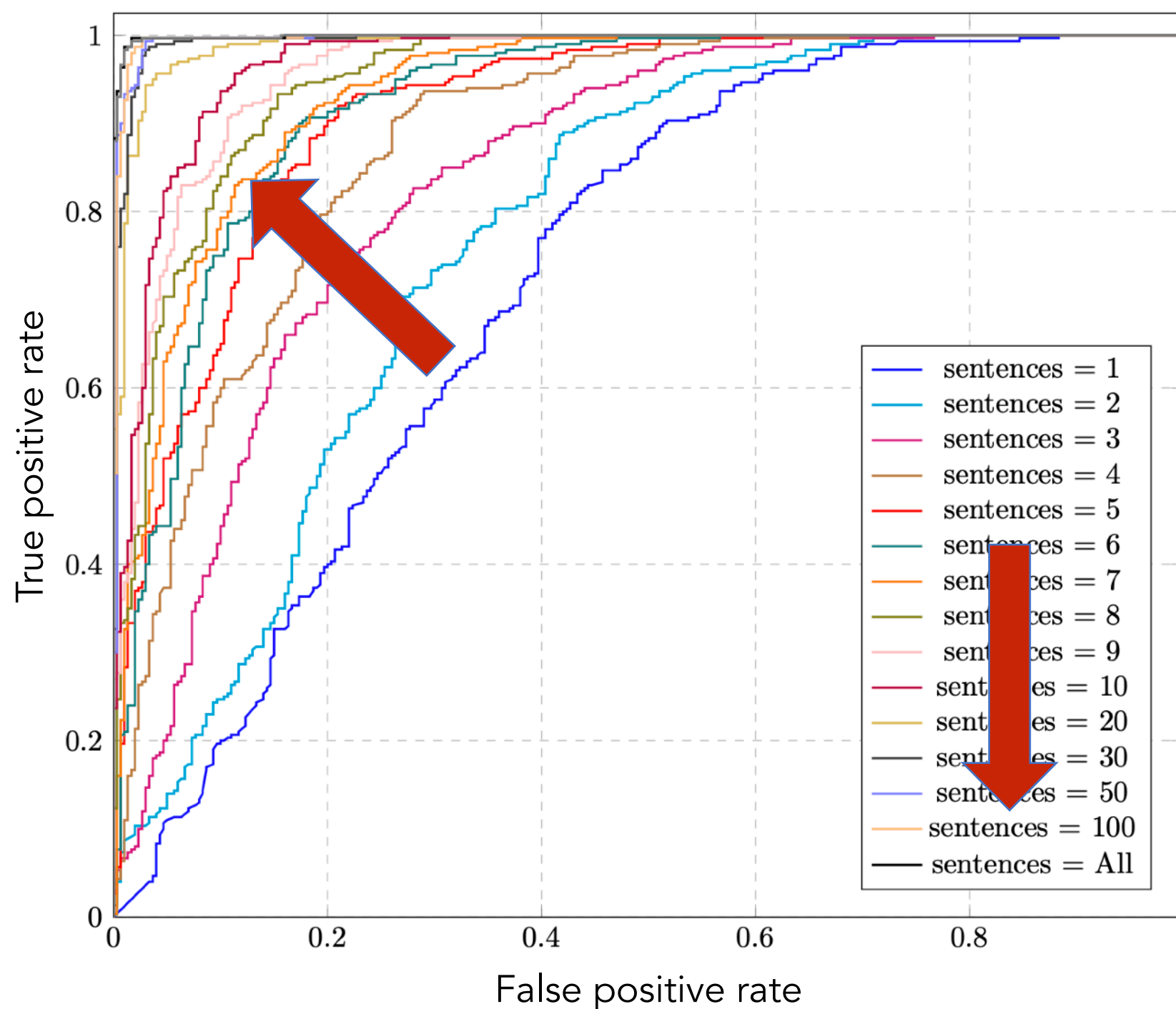


Membership Inference



SATED (Speaker
Annotated TED
talks) dataset

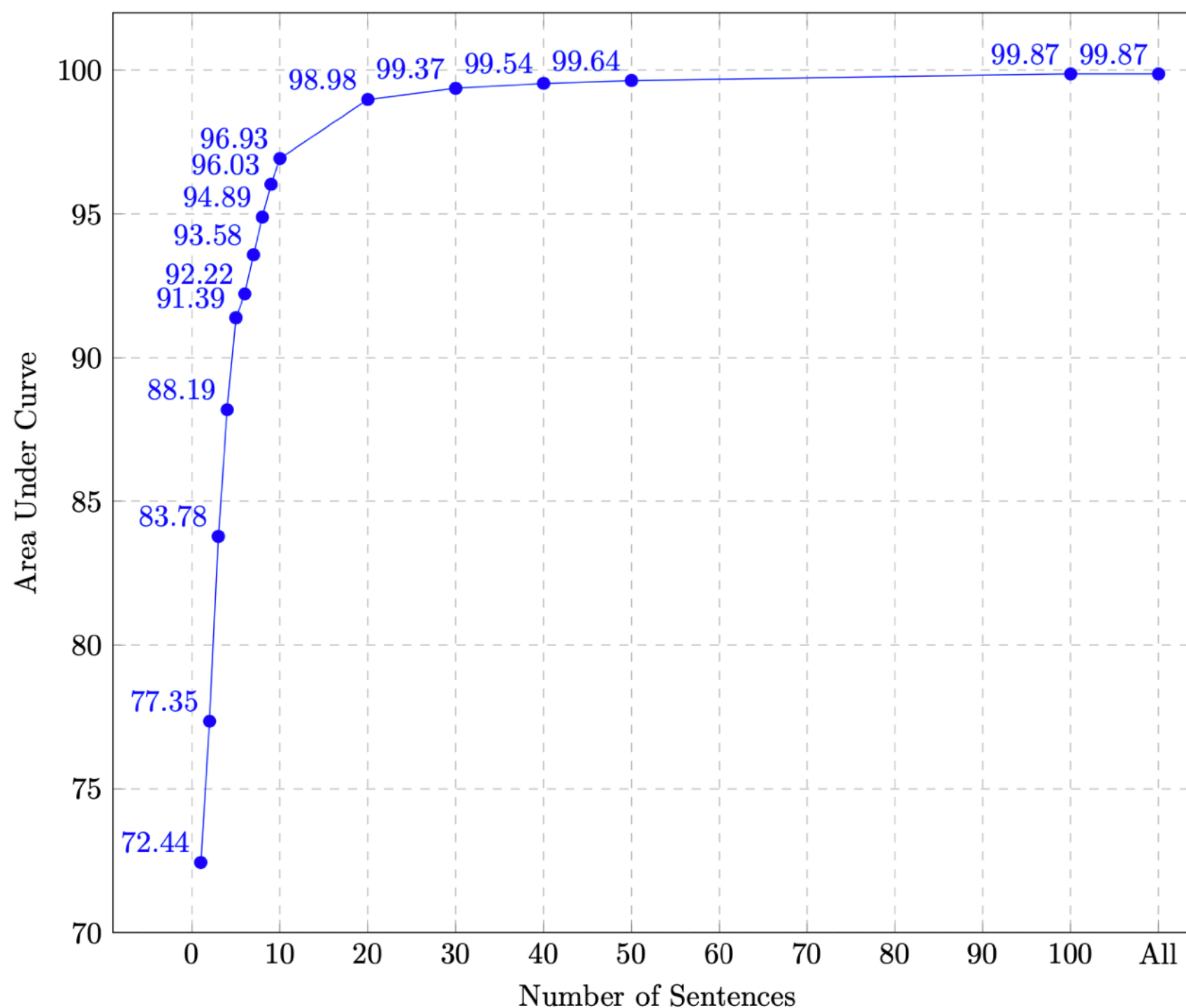
Membership Inference



SATED (Speaker
Annotated TED
talks) dataset

[Maddi] https://github.com/privacytrustlab/ml_privacy_meter based on [Song, Shmatikov] Auditing Data Provenance in Text-Generation Models, KDD'19

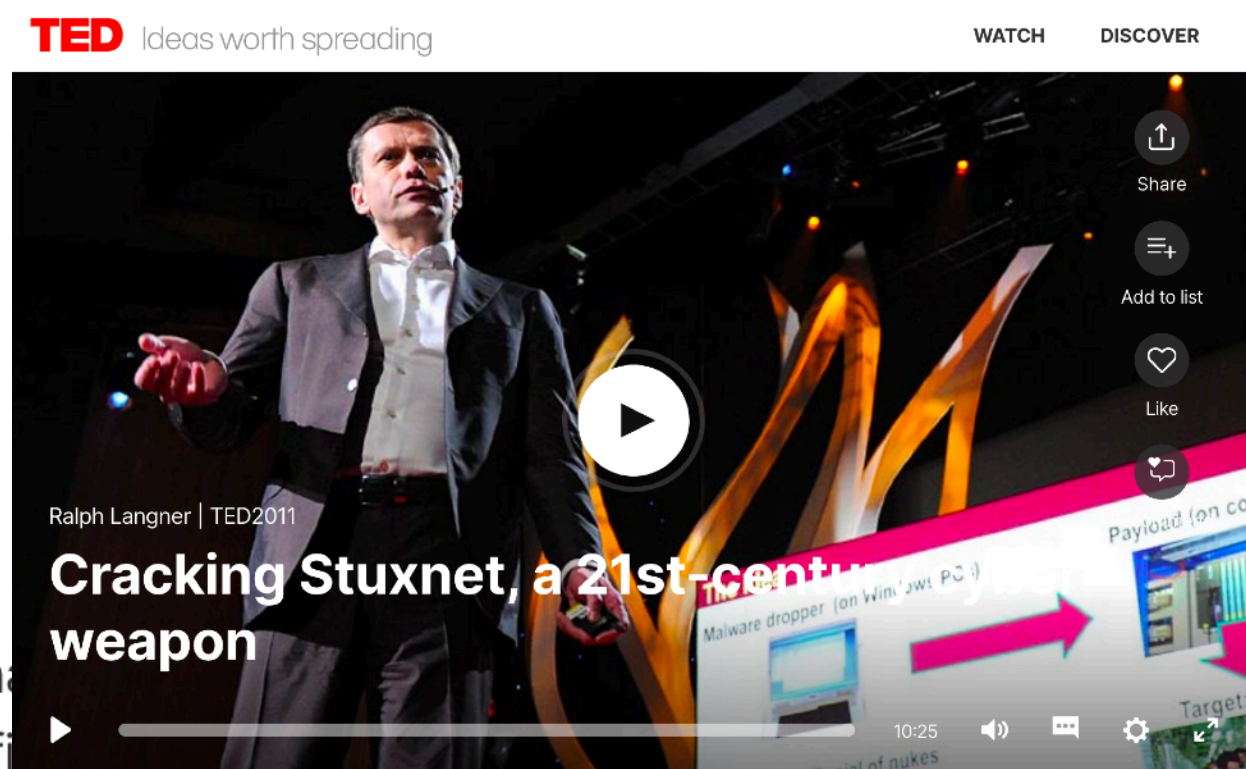
Membership Inference



SATED (Speaker Annotated TED talks) dataset

[Maddi] https://github.com/privacytrustlab/ml_privacy_meter based on [Song, Shmatikov] Auditing Data Provenance in Text-Generation Models, KDD'19

Examples of Vulnerable Training Data

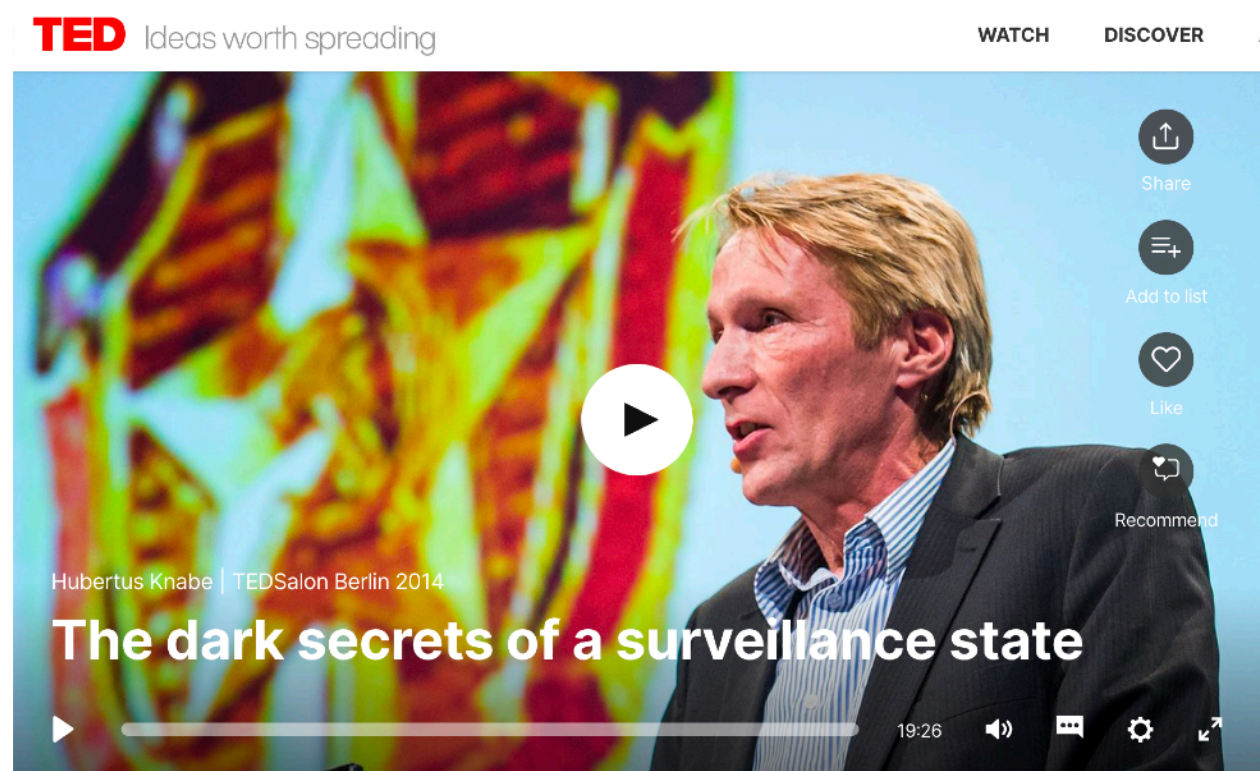


But it gets worse. And this is very important, which is generic. It doesn't have anything to do, in specific, it would work as well, for example, in a power plant or in an automobile factory. It is generic. And you don't have -- as an attacker -- you don't have to deliver this payload by a USB stick, as we saw it in the case of Stuxnet. You could also use conventional worm technology for spreading. Just spread it as

Chris Anderson: I've got a question. Ralph, it's been quite widely reported that people assume that Mossad is the main entity behind this. Is that your opinion?

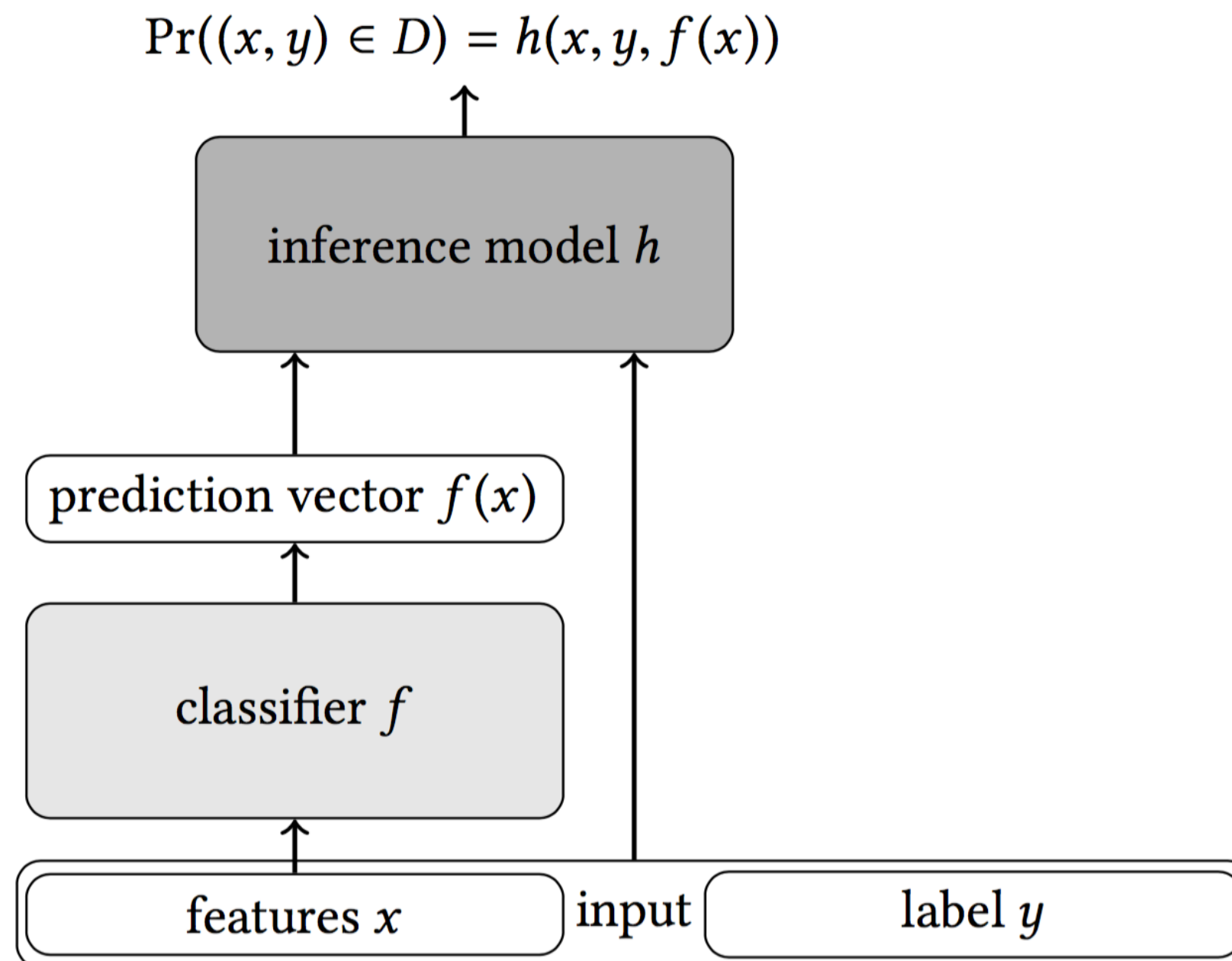
Ralph Langner: Okay, you really want to hear that? Yeah. Okay. My opinion is that the Mossad is involved, but that the leading force is not Israel. So the leading force behind that is the cyber superpower. There is only one, and that's the United States -- fortunately, fortunately. Because otherwise, our problems would even be bigger.

Examples of Vulnerable Training Data

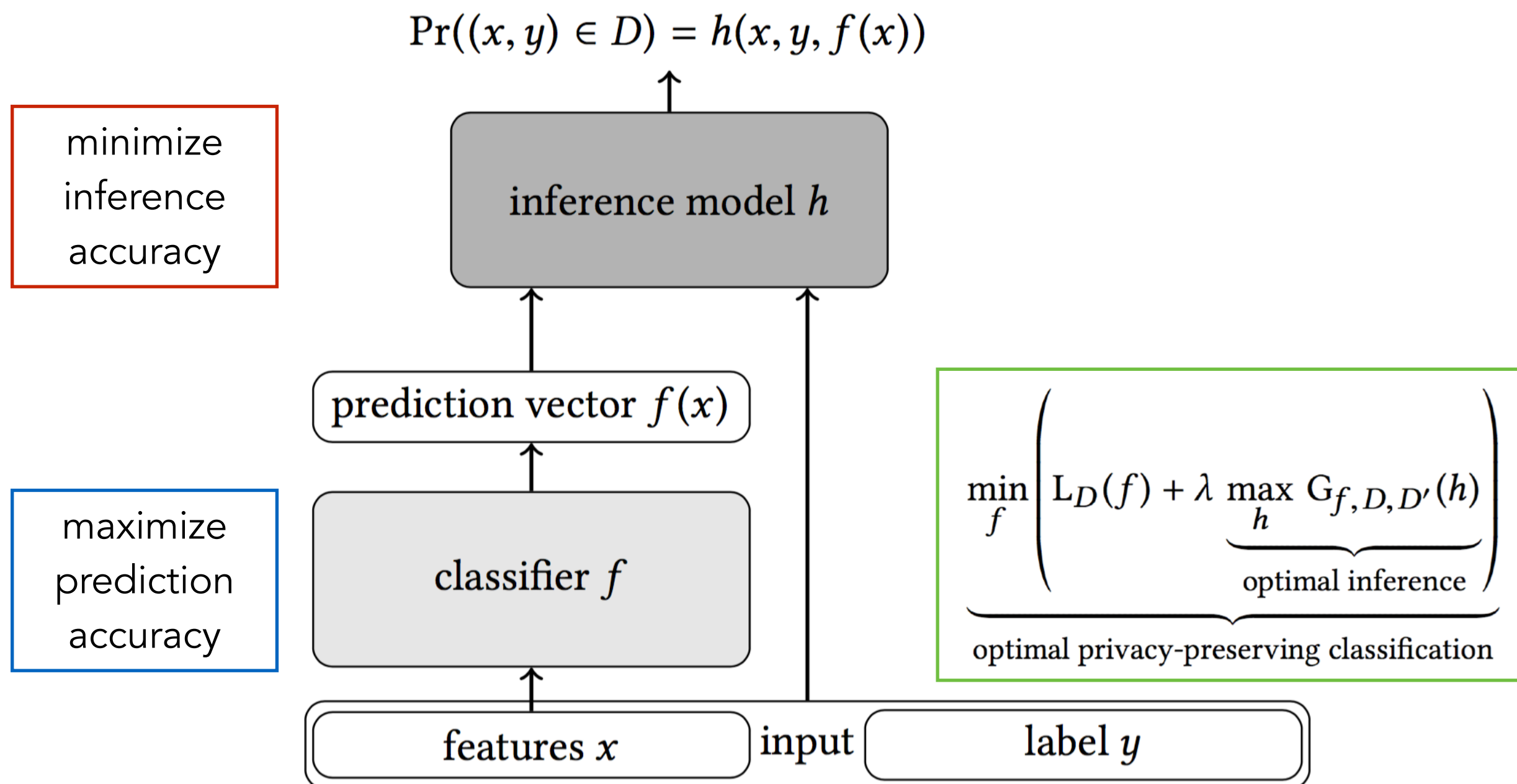


This year, Germany is celebrating the 25th anniversary of the peaceful revolution in East Germany. In 1989, the Communist regime was moved away, the Berlin Wall came down, and one year later, the German Democratic Republic, the GDR, in the East was unified with the Federal Republic of Germany in the West to found today's Germany. Among many other things, Germany inherited the archives of the East German secret police, known as the Stasi. Only two years after its dissolution, its documents were opened to the public, and historians such as me started to study these documents to learn more about how the GDR surveillance state functioned.

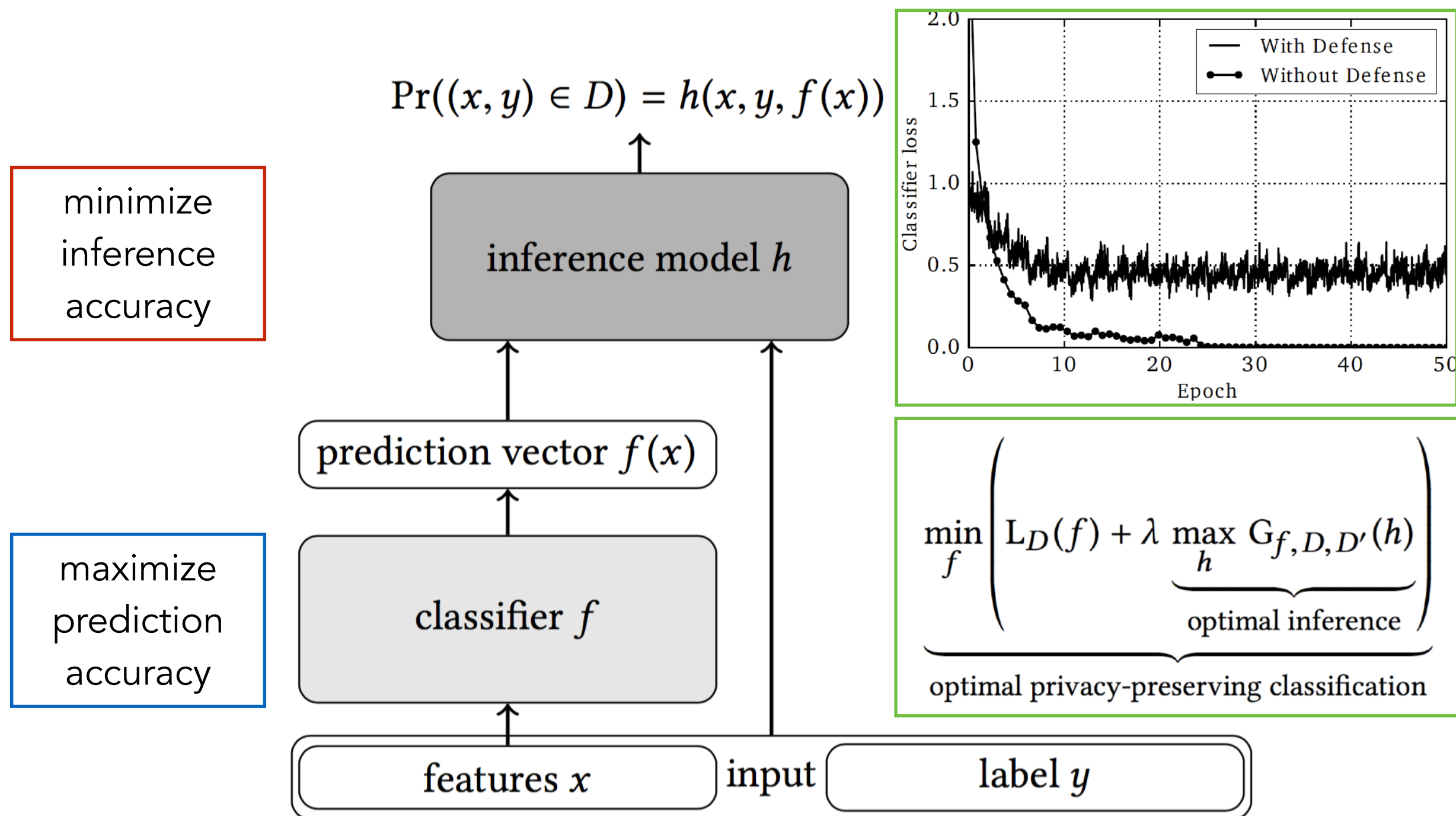
Privacy as a Learning Objective



Privacy as a Learning Objective



Privacy as a Learning Objective

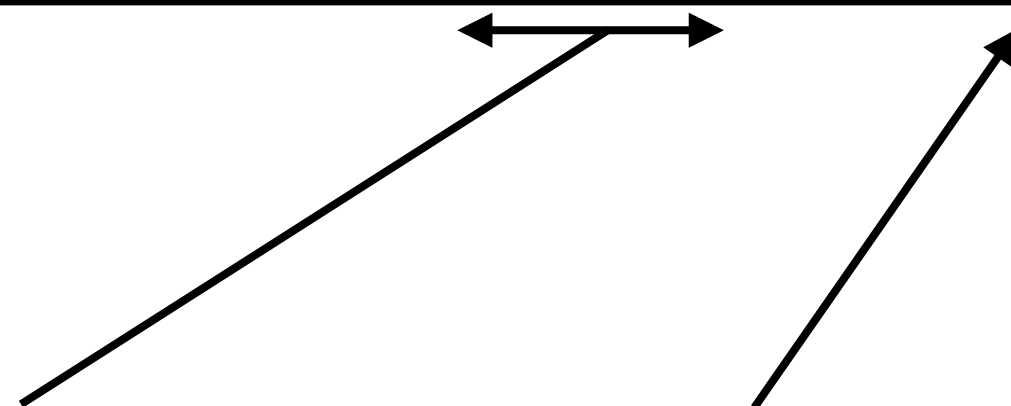


Privacy and Generalization

Dataset	Without defense			With defense		
	Training accuracy	Testing accuracy	Attack accuracy	Training accuracy	Testing accuracy	Attack accuracy
Purchase100	100%	80.1%	67.6%	92.2%	76.5%	51.6%
Texas100	81.6%	51.9%	63%	55%	47.5%	51.0%
CIFAR100- Alexnet	99%	44.7%	53.2%	66.3%	43.6%	50.7%
CIFAR100- DenseNET	100%	70.6%	54.5%	80.3%	67.6%	51.0%

Smaller gap

Random guess



Bound the Worst-case Privacy Loss

- Differential Privacy: Ensure the indistinguishability between two models which are trained on two neighboring datasets.
- Randomize the training algorithm to bound the privacy loss

mechanism (randomized model)

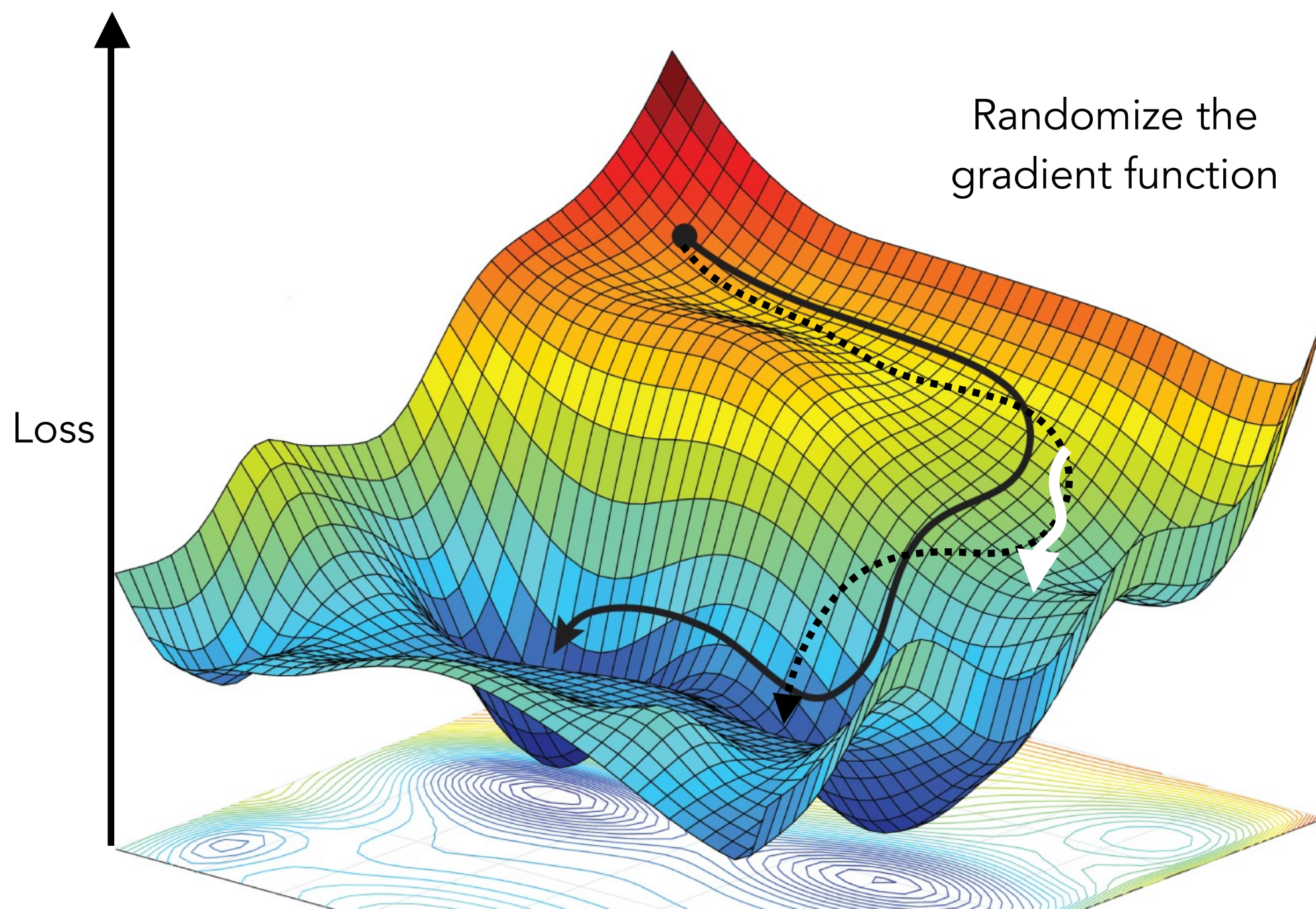
privacy loss

$$\mathcal{L}_{\mathcal{M}(x) \parallel \mathcal{M}(y)}^{(\xi)} = \ln \left(\frac{\Pr[\mathcal{M}(x) = \xi]}{\Pr[\mathcal{M}(y) = \xi]} \right)$$

observables

input (dataset)

DP Stochastic Gradient Descent



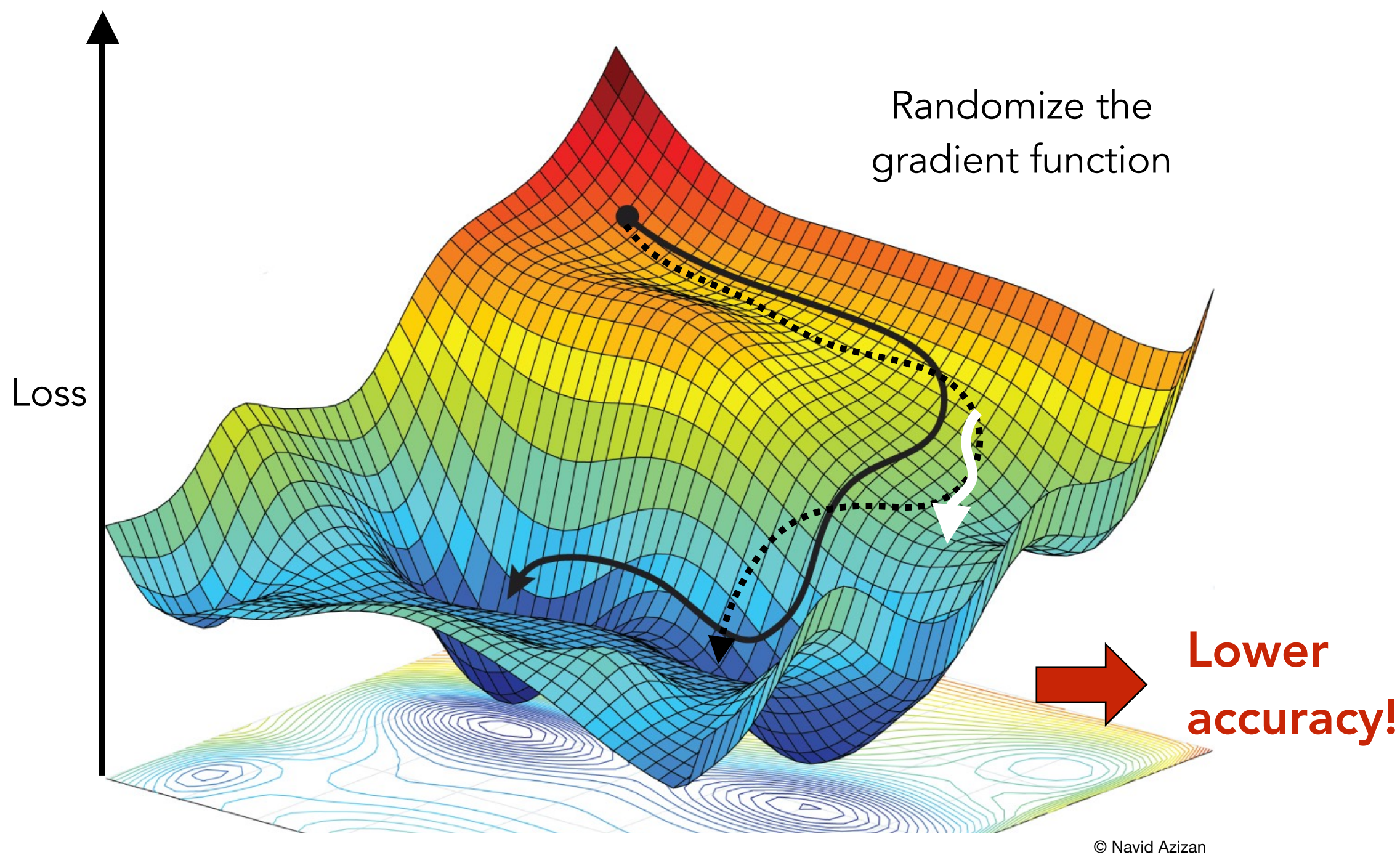
© Navid Azizan

[Bassily, Smith, Thakurta] Private empirical risk minimization: Efficient algorithms and tight error bounds, FOCS'14

[Shokri and Shmatikov] Privacy-Preserving Deep Learning, CCS'15

[Abadi, et al.] Deep learning with differential privacy. CCS'16.

DP Stochastic Gradient Descent



[Bassily, Smith, Thakurta] Private empirical risk minimization: Efficient algorithms and tight error bounds, FOCS'14

[Shokri and Shmatikov] Privacy-Preserving Deep Learning, CCS'15

[Abadi, et al.] Deep learning with differential privacy. CCS'16.

Causes of Performance Loss

- Computation of total privacy loss is not exact (i.e., the upper bound of the privacy loss (epsilon) is not tight). By overestimating the privacy loss, the added noise is larger than what is really needed to achieve the same true level of privacy

Causes of Performance Loss

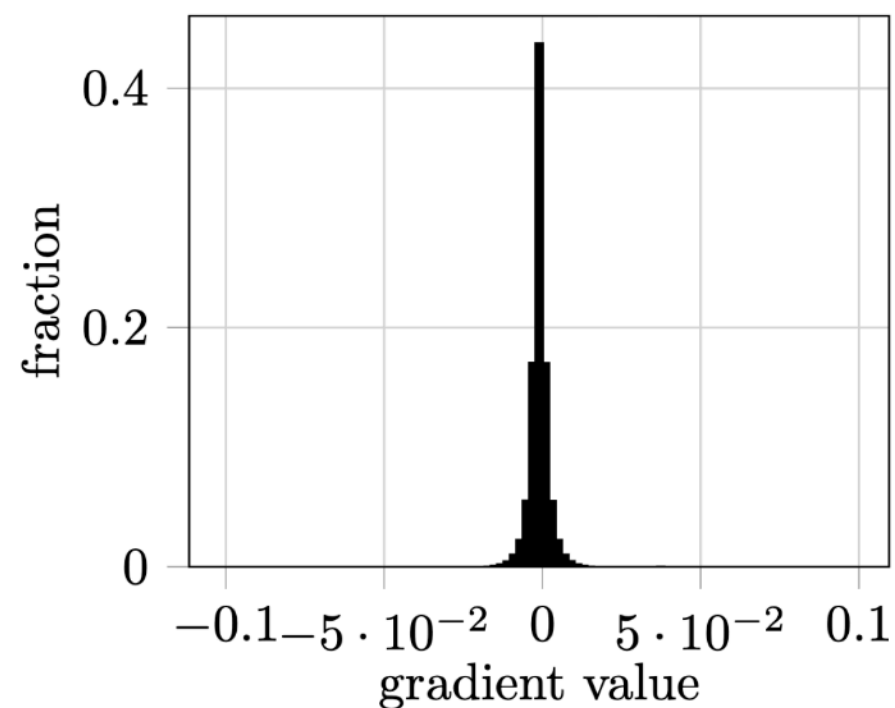
- Computation of total privacy loss is not exact (i.e., the upper bound of the privacy loss (epsilon) is not tight). By overestimating the privacy loss, the added noise is larger than what is really needed to achieve the same true level of privacy
- Gaussian mechanism is not a utility-preserving mechanism for DP SGD

Causes of Performance Loss

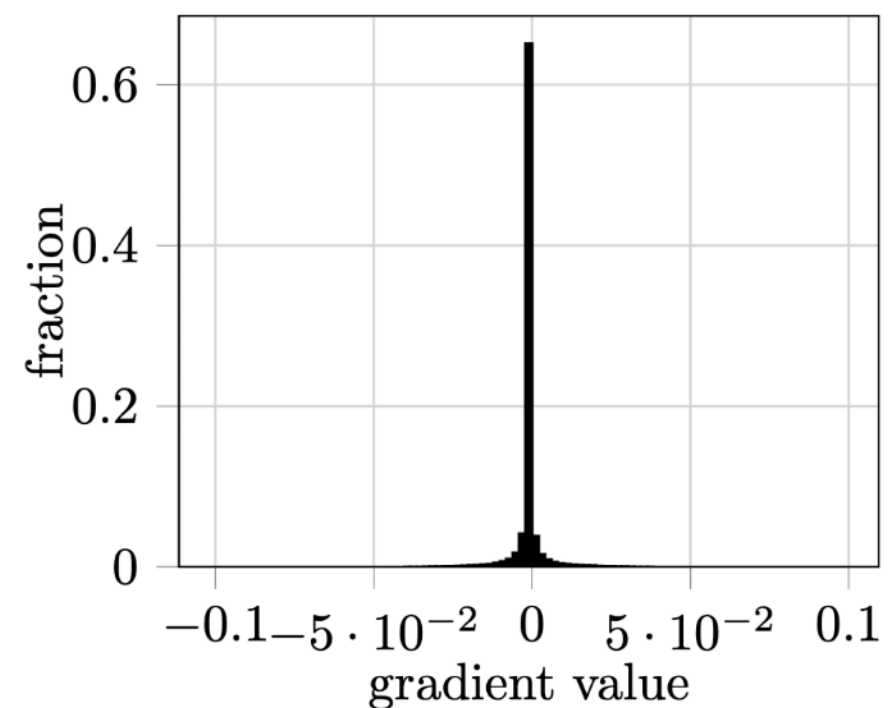
- Computation of total privacy loss is not exact (i.e., the upper bound of the privacy loss (epsilon) is not tight). By overestimating the privacy loss, the added noise is larger than what is really needed to achieve the same true level of privacy
- Gaussian mechanism is not a utility-preserving mechanism for DP SGD
- All randomized gradient vectors are treated equally (but, the signal to noise ratio is not the same across all, and their influence on the parameter vector should not be the same)

Observation

- Gradients follow a symmetric distribution, concentrated around zero



(a) CIFAR



(b) MNIST

- The DP noise would dominate the gradient values

Gradient Coding and De-noising

- Randomize gradients using a student-t distribution
 - To compute DP parameters, encode gradient values into a finite number of samples from a Gaussian distribution

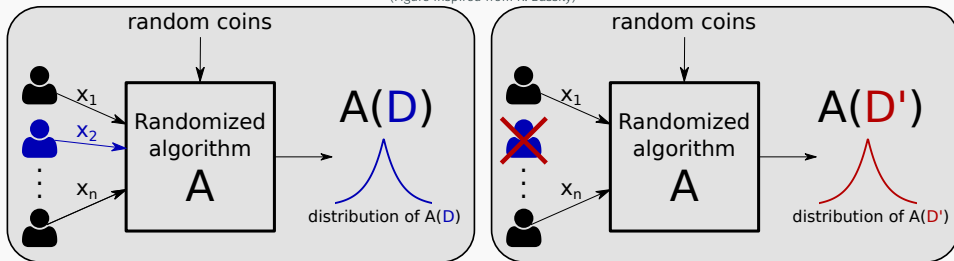
Gradient Coding and De-noising

- Randomize gradients using a student-t distribution
 - To compute DP parameters, encode gradient values into a finite number of samples from a Gaussian distribution
- Weighted update of model parameters
 - Lower the weight if noise dominates the signal

- A bit more formalization

REMINDER: DIFFERENTIAL PRIVACY

(Figure inspired from R. Bassily)



Definition (Differential privacy [Dwork et al., 2006])

Let $\epsilon > 0$ and $\delta \in [0, 1)$. A randomized algorithm $\mathcal{A} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}$ is (ϵ, δ) -differentially private (DP) if for all datasets $D, D' \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|D - D'\|_1 \leq 1$ and for all $\mathcal{S} \subseteq \mathcal{O}$:

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta, \quad (1)$$

where the probability space is over the coin flips of \mathcal{A} .

Definition (Global ℓ_1 sensitivity)

The global ℓ_1 sensitivity of a query (function) $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K$ is

$$\Delta_1(f) = \max_{D, D': \|D - D'\|_1 \leq 1} \|f(D) - f(D')\|_1$$

Definition (Global ℓ_2 sensitivity)

The global ℓ_2 sensitivity of a query (function) $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K$ is

$$\Delta_2(f) = \max_{D, D': \|D - D'\|_1 \leq 1} \|f(D) - f(D')\|_2$$

- How much adding or removing a single record can change the value of the query, measured in ℓ_p norm

Algorithm: Laplace mechanism $\mathcal{A}_{\text{Lap}}(D, f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K, \varepsilon)$

1. Compute $\Delta = \Delta_1(f)$
2. For $k = 1, \dots, K$: draw $Y_k \sim \text{Lap}(\Delta/\varepsilon)$ independently for each k
3. Output $f(D) + Y$, where $Y = (Y_1, \dots, Y_K) \in \mathbb{R}^K$

Theorem (DP guarantees for Laplace mechanism)

Let $\varepsilon > 0$ and $f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K$. The Laplace mechanism $\mathcal{A}_{\text{Lap}}(\cdot, f, \varepsilon)$ satisfies ε -DP.

Algorithm: Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K, \varepsilon, \delta)$

1. Compute $\Delta = \Delta_2(f)$
2. For $k = 1, \dots, K$: draw $Y_k \sim \mathcal{N}(0, \sigma^2)$ independently for each k , where $\sigma = \frac{\sqrt{2 \ln(1.25/\delta)} \Delta}{\varepsilon}$
3. Output $f(D) + Y$, where $Y = (Y_1, \dots, Y_K) \in \mathbb{R}^K$

Theorem (DP guarantees for Gaussian mechanism)

Let $\varepsilon, \delta > 0$ and $f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K$. The Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(\cdot, f, \varepsilon, \delta)$ is (ε, δ) -DP.

THE EXPONENTIAL MECHANISM

LIMITATIONS OF OUTPUT PERTURBATION

- So far we have seen the Laplace and Gaussian mechanisms, which are based on **output perturbation**: $\mathcal{A}(D) = f(D) + Y$
- Can you think of some intrinsic limitations?
- First limitation: they **only work for numeric queries**
- Second limitation: they are useful only if **the utility function is sufficiently regular**

EXAMPLE QUERIES NOT WELL SUITED TO OUTPUT PERTURBATION

- Non-numeric queries
 - What is the most popular website among Firefox users?
 - What is the best set of hyperparameters to train my classifier on the dataset?
- Numeric queries for which two “similar” outputs can have very different utility
 - Which date works better for a set of people to meet?
 - Which price would make the most profit from a set of buyers?

Buyer	Offer
Alice	3€
Bob	4€

- Profit if we set price to 3€: 3€
- Profit if we set price to 3.01€: 3.01€
- Profit if we set price to 4€: 4€
- Profit if we set price to 4.01€: 0€

NON-NUMERIC QUERIES

- We will now consider queries $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}$ with an **abstract output space** \mathcal{O}
 - Example (websites): $\mathcal{O} = \{\text{'Google'}, \text{'Qwant'}, \text{'GitHub'}, \text{'La Quadrature du Net'}, \dots\}$
 - Example (prices): $\mathcal{O} = \{3, 3.01, 4, 4.01, \dots\}$
 - Example (hair color): $\mathcal{O} = \{\text{'dark'}, \text{'blond'}, \text{'brown'}, \text{'red'}\}$
- Associated to \mathcal{O} we have a **score function** (or utility function)

$$s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}$$

- For a dataset $D \in \mathbb{N}^{|\mathcal{X}|}$ and an output $o \in \mathcal{O}$, $s(D, o)$ represents how good it is to **return o when the query is $f(D)$**
- The function s can be arbitrary: it should be designed according to the use-case
- Of course, $o = f(D)$ is usually assigned the maximum score

Definition (Sensitivity of score function)

The sensitivity of a $s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}$ is

$$\Delta(s) = \max_{o \in \mathcal{O}} \max_{D, D': \|D - D'\|_1 \leq 1} |s(D, o) - s(D', o)|$$

- **Worst-case change of score of an output** when adding or removing one record
- Note that sensitivity is only with respect to the dataset (scores can vary arbitrarily across outputs)

Algorithm: Exponential mechanism $\mathcal{A}_{\text{Exp}}(D, f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}, s: \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}, \epsilon)$

1. Compute $\Delta = \Delta(s)$
2. Output $o \in \mathcal{O}$ with probability:

$$\Pr[o] = \frac{\exp\left(\frac{s(D,o) \cdot \epsilon}{2\Delta}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D,o') \cdot \epsilon}{2\Delta}\right)}$$

- Sample $o \in \mathcal{O}$ with **probability proportional to its score** (denominator: normalization)
- Make **high quality outputs exponentially more likely**, at a rate that depends on the sensitivity of the score and the privacy parameter

Theorem (DP guarantees for exponential mechanism)

Let $\epsilon > 0$, $f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}$ and $s: \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}$. $\mathcal{A}_{\text{Exp}}(\cdot, f, s, \epsilon)$ satisfies ϵ -DP.

Proof.

- For clarity, assume \mathcal{O} is finite and let D, D' such that $\|D - D'\|_1 \leq 1$. For any $o \in \mathcal{O}$:

$$\begin{aligned}
 \frac{\Pr[\mathcal{A}_{\text{Exp}}(D, f, S, \varepsilon) = o]}{\Pr[\mathcal{A}_{\text{Exp}}(D', f, S, \varepsilon) = o]} &= \frac{\frac{\exp\left(\frac{s(D, o) \cdot \varepsilon}{2\Delta(S)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D, o') \cdot \varepsilon}{2\Delta(S)}\right)}}{\frac{\exp\left(\frac{s(D', o) \cdot \varepsilon}{2\Delta(S)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D', o') \cdot \varepsilon}{2\Delta(S)}\right)}} = \frac{\exp\left(\frac{s(D, o) \cdot \varepsilon}{2\Delta(S)}\right)}{\exp\left(\frac{s(D', o) \cdot \varepsilon}{2\Delta(S)}\right)} \cdot \frac{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D', o') \cdot \varepsilon}{2\Delta(S)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D, o') \cdot \varepsilon}{2\Delta(S)}\right)} \\
 &= \exp\left(\frac{(s(D, o) - s(D', o))\varepsilon}{2\Delta(S)}\right) \cdot \frac{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D', o') \cdot \varepsilon}{2\Delta(S)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D, o') \cdot \varepsilon}{2\Delta(S)}\right)} \\
 &\leq \exp\left(\frac{\varepsilon}{2}\right) \cdot \exp\left(\frac{\varepsilon}{2}\right) \cdot \frac{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D, o') \cdot \varepsilon}{2\Delta(S)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D, o') \cdot \varepsilon}{2\Delta(S)}\right)} = e^\varepsilon
 \end{aligned}$$



THE EXPONENTIAL MECHANISM: UTILITY GUARANTEES

- Fixing a dataset D , let $s^*(D) = \max_{o \in \mathcal{O}} s(D, o)$
- We show that it is unlikely that \mathcal{A}_{Exp} returns a “bad” output, measured w.r.t. $s^*(D)$

Theorem (Utility guarantees for exponential mechanism)

Let $\varepsilon > 0$, $f: \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ and $s: \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}$. Fix a dataset $D \in \mathbb{N}^{|\mathcal{X}|}$ and let $\mathcal{O}^* = \{o \in \mathcal{O} : s(D, o) = s^*(D)\}$. Then:

$$\Pr \left[s^*(D) - s(\mathcal{A}_{\text{Exp}}(D, f, s, \varepsilon)) \leq \frac{2\Delta(s)}{\varepsilon} \ln \left(\frac{|\mathcal{O}|}{\beta |\mathcal{O}^*|} \right) \right] \geq 1 - \beta$$

- It is highly unlikely that we get utility score smaller than $s^*(D)$ by more than an additive factor of $O((\Delta(s)/\varepsilon) \ln(|\mathcal{O}|))$
- Guarantees are better if several outputs have maximal score (i.e., $|\mathcal{O}^*| \geq 1$)

Proof.

- We want to show that $\Pr[s(\mathcal{A}_{\text{Exp}}(D, f, s, \epsilon)) \leq c] \leq \beta$ for $c = s^*(D) - \frac{2\Delta(s)}{\epsilon} \ln\left(\frac{|\mathcal{O}|}{\beta|\mathcal{O}^*|}\right)$
- Think about “bad” outputs $o \in \mathcal{O}$ with $s(D, o) \leq c$
- Each such o has un-normalized probability mass at most $\exp(\epsilon c / 2\Delta(s))$, hence the entire set has total un-normalized probability mass at most $|\mathcal{O}| \exp(\epsilon c / 2\Delta(s))$
- In contrast, there is at least $|\mathcal{O}^*| \geq 1$ outputs o with $s(D, o) = s^*(D)$, therefore:

$$\begin{aligned} \Pr[s(\mathcal{A}_{\text{Exp}}(D, f, s, \epsilon)) \leq c] &\leq \frac{|\mathcal{O}| \exp(\epsilon c / 2\Delta(s))}{|\mathcal{O}^*| \exp(\epsilon s^*(D) / 2\Delta(s))} \\ &= \frac{|\mathcal{O}|}{|\mathcal{O}^*|} \exp\left(\frac{\epsilon(c - s^*(D))}{2\Delta(s)}\right) \\ &= \beta \end{aligned}$$

□

THE EXPONENTIAL MECHANISM: UTILITY GUARANTEES

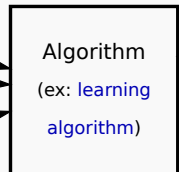
- Let $\mathcal{O} = \{\text{'dark'}, \text{'blond'}, \text{'brown'}, \text{'red'}\}$ and consider the query “What is the most common hair color?” with **counts as scores**
- Suppose that the most common color is 'dark' (with count 500) and the second most common is 'brown' (with count 399)
- For $\varepsilon = 0.1$, what is the probability that \mathcal{A}_{Exp} returns 'dark'?
- Note that $\Delta(s) = 1$, $|\mathcal{O}| = 4$ and $|\mathcal{O}^*| = 1$
- Applying the theorem, we know that the probability of returning an output whose score is larger than $400 = 500 - 20 \ln(4/\beta)$ is at least $1 - \beta$
- This gives $\beta = 4e^{-5}$, hence the probability to get the correct answer is at least $1 - \beta = 0.973$

- The exponential mechanism is the natural building block for answering queries with **arbitrary utilities** and **arbitrary non-numeric range**
- As we have seen, it is often quite easy to analyze
- The set \mathcal{O} of possible outputs should **not be specific to the particular dataset!**
 - Otherwise we violate DP
 - Example of violation: possible prices for items based on actual bids
- The exponential mechanism can define a **complex distribution over an arbitrary large domain**, so it is **not always possible to implement it efficiently**

REMINDER: PRIVATE DATA ANALYSIS

(Figure inspired from R. Bassily)

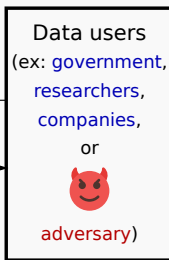
Individuals
(data subjects)



queries

answers

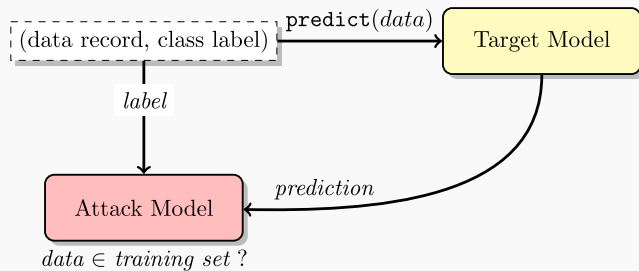
(ex: aggregate statistics,
machine learning model)



- We have focused so far on “simple” aggregate statistics
- How about releasing machine learning models trained on private data?

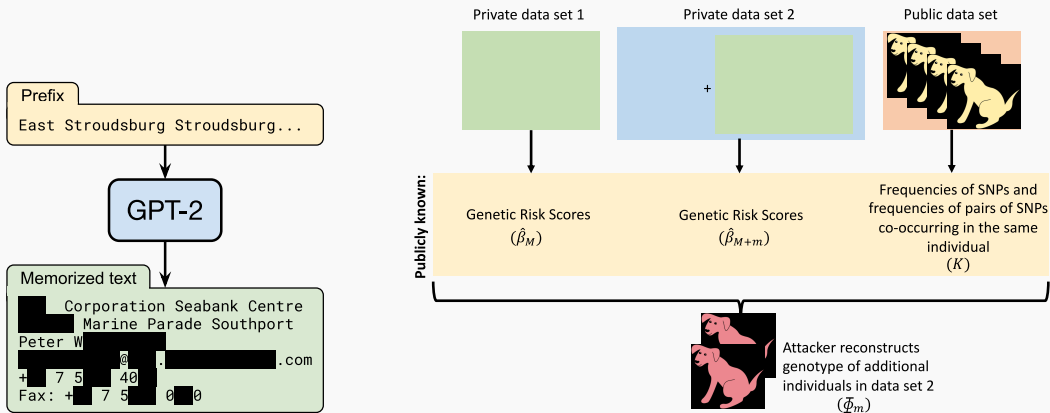
REMINDER: ML MODELS ARE NOT SAFE

- ML models are elaborate kinds of aggregate statistics!
- As such, they are susceptible to **membership inference attacks**, i.e. inferring the presence of a known individual in the training set
- For instance, one can exploit the confidence in model predictions [Shokri et al., 2017] [Carlini et al., 2022]



REMINDER: ML MODELS ARE NOT SAFE

- ML models are also susceptible to **reconstruction attacks**
- For instance, one can **extract sensitive text from large language models** [Carlini et al., 2021] or **run differencing attacks on ML models** [Paige et al., 2020]



1. Reminders on Empirical Risk Minimization (ERM)
2. Private ERM via output perturbation

REMINDERS ON EMPIRICAL RISK MINIMIZATION (ERM)

- For convenience, we focus on supervised learning
- Consider an abstract data space $\mathcal{X} \times \mathcal{Y}$ where \mathcal{X} is the input (feature) space and \mathcal{Y} is the output (label) space
 - For instance, for binary classification with real-valued features: $\mathcal{X} \subset \mathbb{R}^d, \mathcal{Y} = \{-1, 1\}$
- A predictor (model) is a function $h : \mathcal{X} \rightarrow \mathcal{Y}$
- We measure the discrepancy between a prediction $h(x)$ and the true label y using a loss function $L(h; x, y)$

- We have access to a **training set** $D = \{(x_i, y_i)\}_{i=1}^n$ of n data points
- Each data point (x_i, y_i) is assumed to be **drawn independently from a fixed but unknown distribution** μ
- The goal of ML is to find a predictor h with small **expected risk**:

$$R(h) = \mathbb{E}_{(x,y) \sim \mu} [L(h; x, y)]$$

- Since μ is unknown, we will use the training set to construct a proxy to R

EMPIRICAL RISK MINIMIZATION (ERM)

- We thus define the **empirical risk**:

$$\hat{R}(h; D) = \frac{1}{n} \sum_{i=1}^n L(h; x_i, y_i)$$

- Assume that we work with predictors $h_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ **parameterized by** $\theta \in \Theta \subseteq \mathbb{R}^p$
- For notational convenience, we use $L(\theta; x, y)$, $R(\theta)$ and $\hat{R}(\theta)$ to denote $L(h_\theta; x, y)$, $R(h_\theta; D)$ and $\hat{R}(h_\theta; D)$, and omit the dependency on D when it is clear from the context
- **Empirical Risk Minimization** (ERM) consists in choosing the parameters

$$\hat{\theta} \in \arg \min_{\theta \in \Theta} [F(\theta; D) := \hat{R}(\theta; D) + \lambda \psi(\theta)]$$

- ψ is a **regularizer** and $\lambda \geq 0$ a trade-off parameter

USEFUL PROPERTIES

- We typically work with loss functions that are **differentiable in θ** : for $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we denote the gradient of L at θ by $\nabla L(\theta; x, y) \in \mathbb{R}^p$
- We also like the loss function, its gradient and/or the regularizer to be **Lipschitz**

Definition (Lipschitz function)

Let $l > 0$. A function f is l -Lipschitz with respect to some norm $\|\cdot\|$ if for all $\theta, \theta' \in \Theta$:

$$|f(\theta) - f(\theta')| \leq l \|\theta - \theta'\|.$$

If f is differentiable and $\|\cdot\| = \|\cdot\|_2$, the above property is equivalent to:

$$\|\nabla f(\theta)\|_2 \leq l, \quad \forall \theta \in \Theta.$$

- It is also useful when the loss and/or regularizer are **convex** or **strongly convex**

Definition (Strongly convex function)

Let $s \geq 0$. A differentiable function f is s -strongly convex if for all $\theta, \theta' \in \Theta$:

$$f(\theta') \geq f(\theta) + \nabla f(\theta)^\top (\theta - \theta') + \frac{s}{2} \|\theta - \theta'\|_2^2,$$

or equivalently:

$$(\nabla f(\theta) - \nabla f(\theta'))^\top (\theta - \theta') \geq s \|\theta - \theta'\|_2^2,$$

For $s = 0$, we simply say that f is convex.

EXAMPLE: LOGISTIC REGRESSION

- Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$
- Pick a family of **linear models** $h_\theta(x) = \text{sign}[\theta^\top x + b]$ for $\theta \in \Theta = \mathbb{R}^p$
- Pick the **logistic loss** $L(\theta; x, y) = \log(1 + e^{-y(\theta^\top x + b)})$, which is **$\|x\|$ -Lipschitz** and **convex**
- For $\psi(\theta) = 0$, the ERM problem gives **logistic regression**
- If we additionally set $\psi(\theta) = \|\theta\|_2^2$, we obtain **ℓ_2 -regularized logistic regression**
- Then $\psi(\theta)$ is **2-strongly convex** and $F(\theta) = \hat{R}(\theta) + \lambda\psi(\theta)$ is **2λ -strongly convex**

PRIVATE ERM VIA OUTPUT PERTURBATION

- We would like to privately release a model trained on private data
- A differentially private machine learning algorithm $\mathcal{A} : \mathbb{N}^{|\mathcal{X} \times \mathcal{Y}|} \rightarrow \Theta$ should guarantee that for all neighboring datasets D, D' and for all $S_\Theta \subseteq \Theta$:

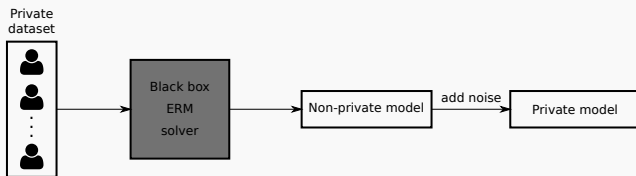
$$\Pr[\mathcal{A}(D) \in S_\Theta] \leq e^\epsilon \Pr[\mathcal{A}(D') \in S_\Theta] + \delta$$

- **Important note:** in ML, we consider a slightly different neighboring relation where two neighboring datasets $D, D' \in (\mathcal{X} \times \mathcal{Y})^n$ have same size n and differ on one record
 - This corresponds to replacing instead adding/removing one record
 - This is for convenience: normalization term in empirical risk is $1/n$ for both D and D'

- Does DP seem compatible with the objective of ML?
- **Yes!** Intuitively, a model which does not change too much when trained on datasets that differ by a single point should **generalize well** (because it **does not overfit**)
- This is related to the notion of **algorithmic stability** [Bousquet and Elisseeff, 2002], which is known to be a sufficient condition for generalization
- There are formal connections between DP and algorithmic stability [Wang et al., 2016]: in particular, “DP implies stability”

DIFFERENTIALLY PRIVATE ERM VIA OUTPUT PERTURBATION

- ERM is a more complicated kind of “query” than those we have seen so far
- Still, can we re-use some ideas to construct DP-ERM algorithms?
- A natural approach is to rely on **output perturbation**:



Formally: $\mathcal{A}(D) = \hat{\theta} + \eta$, where $\hat{\theta} \in \arg \min_{\theta \in \Theta} [F(\theta; D) := \hat{R}(\theta; D) + \lambda\psi(\theta)]$

- To calibrate the noise, we need to **bound the sensitivity of $\hat{\theta}$**
- In some cases, this sensitivity may actually be quite high!
 - Non-regularized objectives with expressive models (e.g., deep neural networks)
 - ℓ_1 -regularized models such as LASSO, which are known to be unstable [Xu et al., 2012]

Theorem (ℓ_2 sensitivity for ERM [Chaudhuri et al., 2011])

Let $\Theta = \mathbb{R}^p$. If the regularizer ψ is differentiable and 1-strongly convex, and the loss function $L(\cdot; x, y)$ is convex, differentiable and 1-Lipschitz w.r.t. the ℓ_2 norm for all $x, y \in \mathcal{X} \times \mathcal{Y}$, then the ℓ_2 sensitivity of $\arg \min_{\theta} F(\theta)$ is at most $2/n\lambda$.

- As expected, sensitivity decreases with n (the size of the dataset)
- Weak regularization leads to large upper bound on sensitivity
- Let's prove this theorem!

SENSITIVITY BOUND FOR SOME REGULARIZED ERM FORMULATIONS

Lemma

Let $G(\theta)$ and $g(\theta)$ be two vector-valued functions that are continuous and differentiable everywhere. Assume that $G(\theta)$ and $G(\theta) + g(\theta)$ are λ -strongly convex.

If $\theta_1 = \arg \min_{\theta} G(\theta)$ and $\theta_2 = \arg \min_{\theta} G(\theta) + g(\theta)$, then $\|\theta_1 - \theta_2\|_2 \leq \frac{1}{\lambda} \max_{\theta} \|\nabla g(\theta)\|_2$.

Proof.

- By the optimality of θ_1 and θ_2 , we have $\nabla G(\theta_1) = \nabla G(\theta_2) + \nabla g(\theta_2) = 0$
- As $G(\theta)$ is strongly convex, we have $(\nabla G(\theta_1) - \nabla G(\theta_2))^{\top} (\theta_1 - \theta_2) \geq \lambda \|\theta_1 - \theta_2\|_2^2$
- Using Cauchy-Schwartz inequality and the above two results, we obtain:

$$\|\theta_1 - \theta_2\|_2 \|\nabla g(\theta_2)\|_2 \geq (\theta_1 - \theta_2)^{\top} \nabla g(\theta_2) = (\nabla G(\theta_1) - \nabla G(\theta_2))^{\top} (\theta_1 - \theta_2) \geq \lambda \|\theta_1 - \theta_2\|_2^2$$

- Dividing both sides by $\lambda \|\theta_1 - \theta_2\|$ gives us the result



Proof of the theorem.

- Let $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, $D' = \{(x'_1, y'_1), \dots, (x_n, y_n)\}$ be two neighboring datasets that differ only in their first point
- Denoting $\hat{\theta} = \arg \min_{\theta} F(\theta; D)$ and $\hat{\theta}' = \arg \min_{\theta} F(\theta; D')$, we want to bound $\|\hat{\theta} - \hat{\theta}'\|$
- We define a convenient differentiable function

$$g(\theta) = F(\theta; D') - F(\theta; D) = \frac{1}{n} \left(L(\theta; x'_1, y'_1) - L(\theta; x_1, y_1) \right)$$

- By using the 1-Lipschitz property of L we have for any θ :

$$\|\nabla g(\theta)\| = \left\| \frac{1}{n} \left(\nabla L(\theta; x'_1, y'_1) - \nabla L(\theta; x_1, y_1) \right) \right\| \leq \frac{2}{n}$$

□

Proof of the theorem.

- To complete the proof, we will show that $\|\hat{\theta} - \hat{\theta}'\| \leq \frac{1}{\lambda} \max_{\theta} \|\nabla g(\theta)\|$
- Let $G(\theta) = F(\theta; D)$ and recall the definition of $g(\theta) = F(\theta; D') - F(\theta; D)$
- Since L is convex and ψ is 1-strongly convex, $G(\theta)$ and $G(\theta) + g(\theta) = F(\theta; D')$ are λ -strongly convex (as well as differentiable)
- Furthermore, $\hat{\theta}$ and $\hat{\theta}'$ are their corresponding minimizers
- Hence we can apply the lemma, which gives us the desired result



Algorithm: DP-ERM via output perturbation $\mathcal{A}_{\text{DP-ERM}}(D, L, \psi, \lambda, \varepsilon, \delta)$

1. Compute ERM solution $\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} F(\theta)$
2. For $j = 1, \dots, p$: draw $Y_j \sim \mathcal{N}(0, \sigma^2)$ independently for each j , where $\sigma = \frac{2\sqrt{2 \ln(1.25/\delta)}}{n\lambda\varepsilon}$
3. Output $\hat{\theta} + Y$, where $Y = (Y_1, \dots, Y_p) \in \mathbb{R}^p$

Theorem (DP guarantees for DP-ERM via output perturbation)

Let $\varepsilon, \delta > 0$ and $\Theta = \mathbb{R}^p$. Let the loss function L and the regularizer ψ satisfy the conditions of the previous theorem. Then $\mathcal{A}_{\text{DP-ERM}}(\cdot, L, \psi, \varepsilon, \delta)$ is (ε, δ) -DP.

- Proof: a direct application of the **Gaussian mechanism** with the previous theorem

DP-ERM VIA OUTPUT PERTUBATION: UTILITY GUARANTEES

- Utility is the **excess (empirical or expected) risk w.r.t. the non-private solution**

Theorem (Utility guarantees for DP-ERM via output perturbation [Chaudhuri et al., 2011])

Consider linear models with $L(\theta; x, y) := L(\theta^\top x, y)$ and normalized data such that $\|x\|_2 \leq 1$ for all $x \in \mathcal{X}$. Let $\psi(\theta) = \frac{1}{2}\|\theta\|_2^2$, $\gamma > 0$ and $\beta > 0$. Let L be differentiable and 1-Lipschitz w.r.t. the ℓ_2 norm and ∇L be 1-Lipschitz w.r.t. the ℓ_1 norm. Let $\theta^* \in \arg \min R(\theta)$ be a minimizer of the expected risk. If n is of order

$$O\left(\max\left(\frac{\|\theta^*\|_2^2 \log(\frac{1}{\beta})}{\gamma^2}, \frac{p \log(\frac{p}{\beta}) \|\theta^*\|_2 \sqrt{\log(\frac{1}{\delta})}}{\gamma \varepsilon}, \frac{p \log(\frac{p}{\beta}) \|\theta^*\|_2^2 \sqrt{\log(\frac{1}{\delta})}}{\gamma^{3/2} \varepsilon}\right)\right),$$

then the output θ_{priv} of $\mathcal{A}_{\text{DP-ERM}}$ satisfies $\Pr[R(\theta_{\text{priv}}) \leq R(\theta^*) + \gamma] \geq 1 - 2\beta$.

- The first term in the max is the sample size needed for non-private ERM
- This theorem shows that DP-ERM via output perturbation is well-founded: it **matches the utility of the non-private case at the cost of a larger training set**

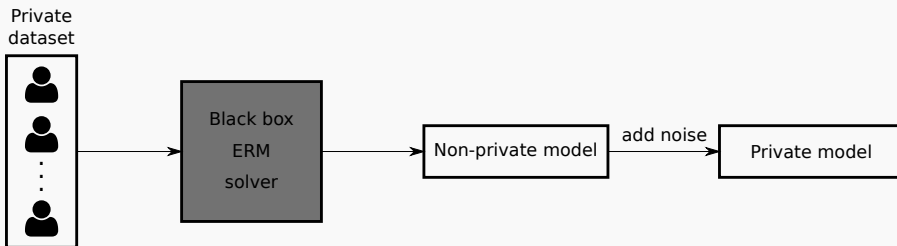
- An advantage of DP-ERM via output perturbation is that it is **simple to implement** on top of non-private algorithms
- However it requires **restrictive assumptions on the loss function and regularizer**
- In practice, **ERM is not solved exactly** but only to a certain precision using iterative solvers like (stochastic) gradient descent
- **Approximate solutions may have small sensitivity**, even if no (strongly convex) regularization is used [Zhang et al., 2017]

1. **Objective perturbation** [Chaudhuri et al., 2011]: output the solution to ERM with a perturbed objective (not covered in the lectures)
2. **Gradient perturbation** [Bassily et al., 2014, Abadi et al., 2016]: perturb the gradients of a gradient-based algorithm (**next lecture!**)

DIFFERENTIALLY PRIVATE SGD

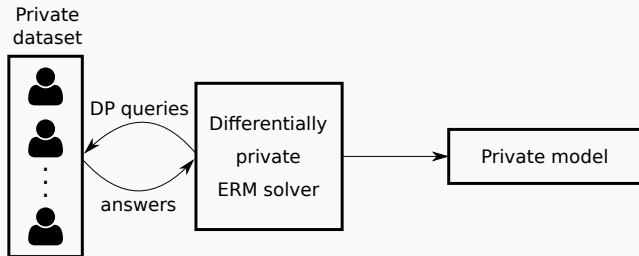
LIMITATIONS OF DP-ERM VIA OUTPUT PERTURBATION

1. It requires **restrictive assumptions** on the loss function and regularizer
2. The sensitivity is likely to be **pessimistic** as it **treats ERM as a black box**



ALTERNATIVE APPROACH: DIFFERENTIALLY PRIVATE ERM SOLVER

- Another approach is to **design differentially private ERM solvers**
- Such a solver (optimization algorithm) must **interact with the data only through DP mechanisms**
- The idea is to perturb only the quantities accessed by a particular solver



NON-PRIVATE STOCHASTIC GRADIENT DESCENT (SGD)

- For simplicity, let us assume that $\psi(\theta) = 0$ (no regularization)
- Denote by $\Pi_{\Theta}(\theta) = \arg \min_{\theta' \in \Theta} \|\theta - \theta'\|_2$ the projection operator onto Θ

Algorithm: Non-private (projected) SGD

- Initialize parameters to $\theta^{(0)} \in \Theta$
 - For $t = 0, \dots, T - 1$:
 - Pick $i_t \in \{1, \dots, n\}$ uniformly at random
 - $\theta^{(t+1)} \leftarrow \Pi_{\Theta}(\theta^{(t)} - \gamma_t \nabla L(\theta^{(t)}; x_{i_t}, y_{i_t}))$
 - Return $\theta^{(T)}$
-
- SGD is a **natural candidate solver**: simple, flexible, scalable, heavily used in ML
 - How to design a DP version of SGD?

- We have already seen ingredients to do this in previous lectures
- Assume that $L(\cdot; x, y)$ is l -Lipschitz with respect to the ℓ_2 norm for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$
- Then we know that for all x, y, θ we have $\|\nabla L(\theta; x, y)\| \leq l$
- Therefore, at any step t of SGD, the ℓ_2 sensitivity of individual gradients is bounded:

$$\sup_{x, y, x', y'} \|\nabla L(\theta; x, y) - \nabla L(\theta; x', y')\| \leq 2l, \quad \forall \theta \in \Theta$$

and we can use the Gaussian mechanism

- It feels like we can do better...

Algorithm: Differentially Private SGD $\mathcal{A}_{\text{DP-SGD}}(D, L, \varepsilon, \delta)$

- Initialize parameters to $\theta^{(0)} \in \Theta$ (must be independent of D)
- For $t = 0, \dots, T - 1$:
 - Pick $i_t \in \{1, \dots, n\}$ uniformly at random
 - $\eta^{(t)} \leftarrow (\eta_1^{(t)}, \dots, \eta_p^{(t)}) \in \mathbb{R}^p$ where each $\eta_j^{(t)} \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \frac{16L\sqrt{T\ln(2/\delta)\ln(2.5T/\delta n)}}{n\varepsilon}$
 - $\theta^{(t+1)} \leftarrow \Pi_{\Theta}(\theta^{(t)} - \gamma_t(\nabla L(\theta^{(t)}; x_{i_t}, y_{i_t}) + \eta^{(t)}))$
- Return $\theta^{(T)}$

- **More data** (larger n) \rightarrow **less noise** added to each gradient
- **More iterations** (larger T) \rightarrow **more noise** added to each gradient

Theorem (DP guarantees for DP-SGD)

Let $\varepsilon \leq 1, \delta > 0$. Let the loss function $L(\cdot; x, y)$ be l -Lipschitz w.r.t. the ℓ_2 norm for all $x, y \in \mathcal{X} \times \mathcal{Y}$. Then $\mathcal{A}_{\text{DP-SGD}}(\cdot, L, \varepsilon, \delta)$ is (ε, δ) -DP.