EPFL

SPRING

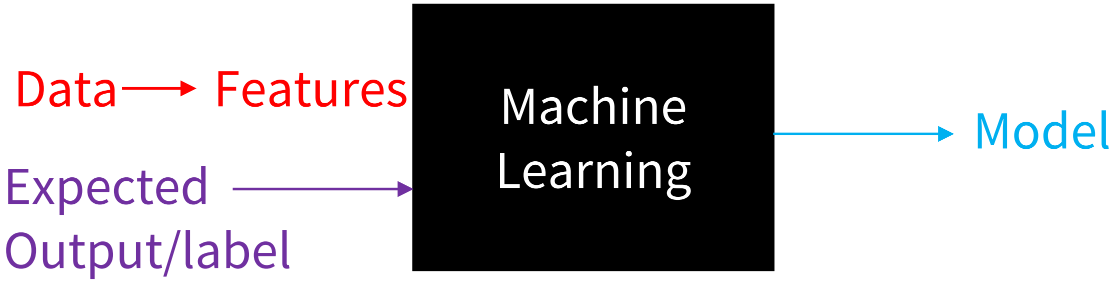
SECURITY AND PRIVACY ENGINEERING LABORATORY

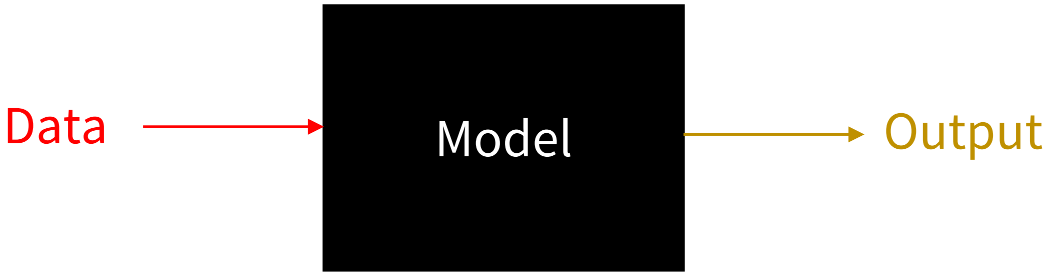
Machine Learning
Adversaries, attacks, and mitigations

Carmela Troncoso

Slides adapted from Rebekah Overdorf, Bogdan Kulynych, Dario Pasquini and Theresa Stadler

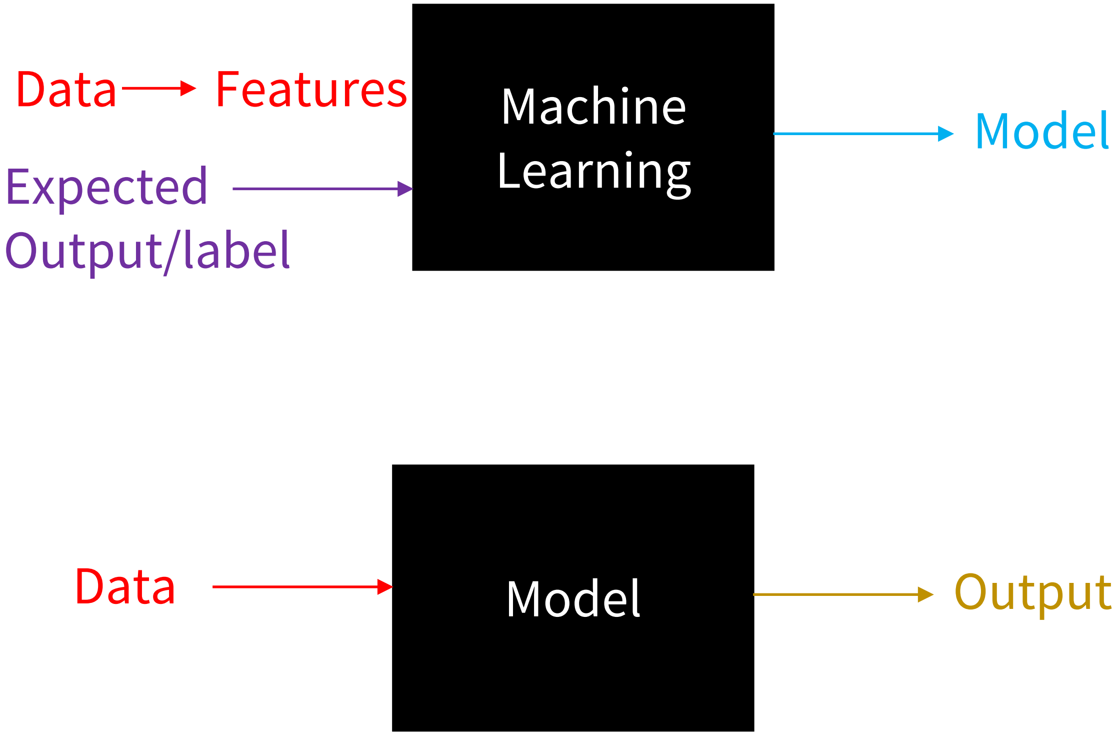
Machine Learning – Privacy concerns?





1. **Data Collection**: Get data that you know something about
2. **Train the Model**: "Teach" the machine about that data
3. **Test the model**: "Check" that the machine on something you know
4. **Deploy**: Make the model available to someone else (e.g., via APIs)

Machine Learning – Privacy concerns?



1. **Data Collection**: Get data that you know something about
2. **Train the Model**: “Teach” the machine about that data
3. **Test the model**: “Check” that the machine on something you know
4. **Deploy**: Make the model available to someone else (e.g., via APIs)

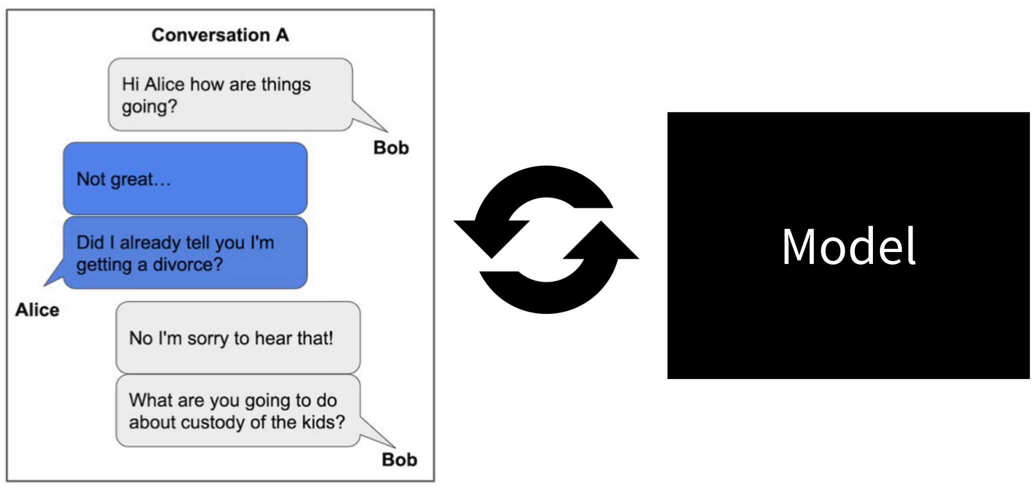
Machine Learning – Privacy concerns?

Our main concern is the privacy of the training set. Why?:

- 🔒 Training data may be sensitive!
- 💸 Training data is expensive to collect.

Training phase:

Training instance:



Conversation A

Hi Alice how are things going?

Bob

Not great...

Did I already tell you I'm getting a divorce?

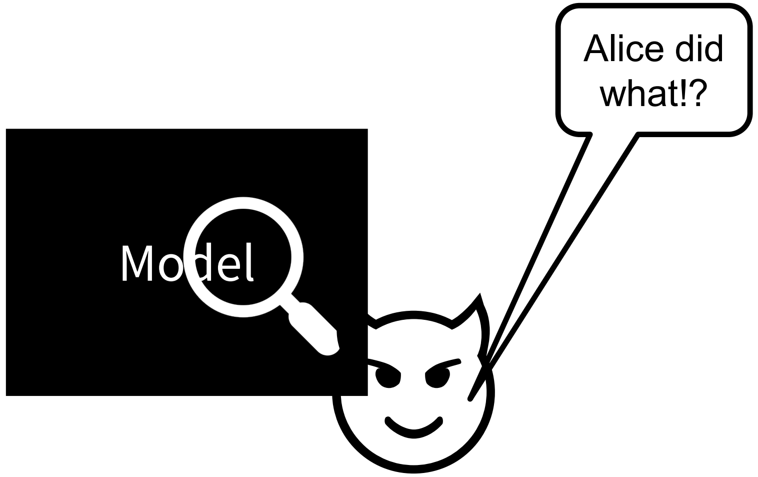
Alice

No I'm sorry to hear that!

What are you going to do about custody of the kids?

Bob

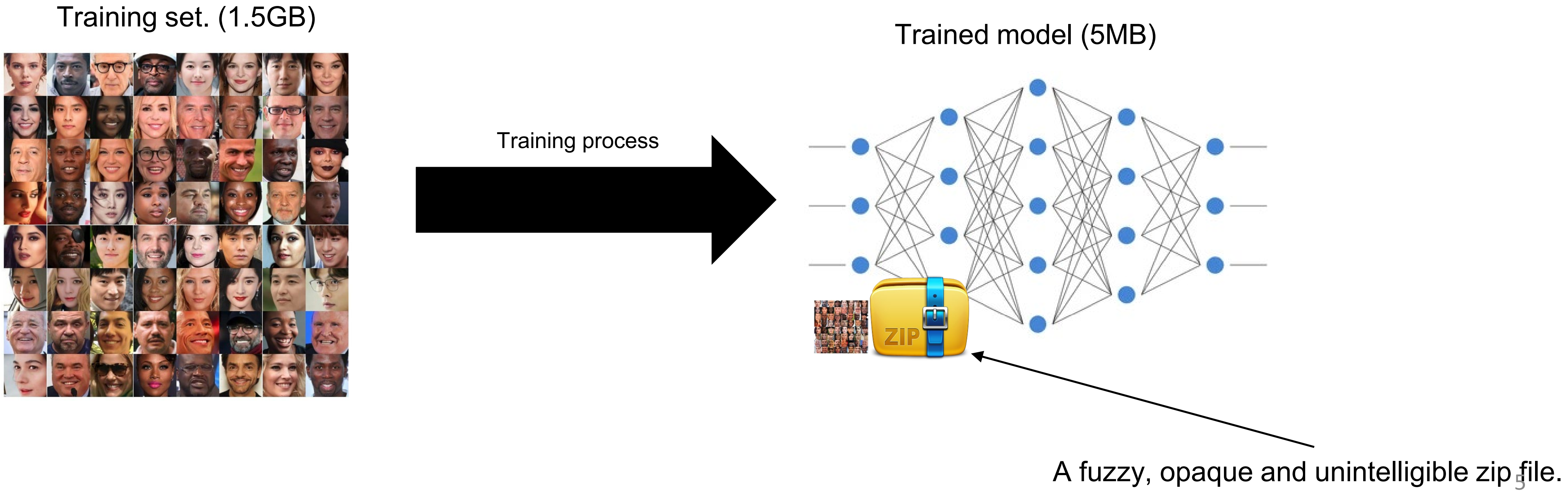
Deployment phase:



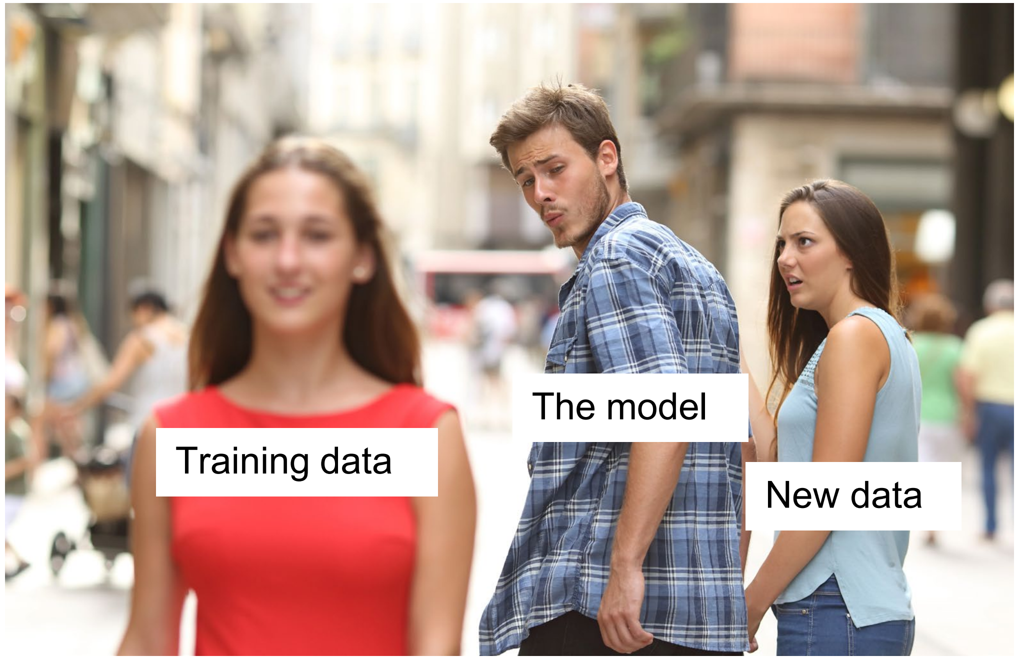
Machine Learning and Privacy: The underlying problem

"A trained machine learning model is just a lossy compressed version of the training set."

Every security-person since the machine learning era.



Privacy of the training set: The foundations



Intuitions:

- A trained model **behaves differently** whether applied on the training data or unseen data points.
- **Privacy attacks are just about characterizing and exploiting* those behaviors** in order to infer information about the training set *(or inducing them in the active security model).

What can an adversary learn from model outputs?

What can an adversary do with outputs?

- **Membership inference**: say whether a sample was **in** the training set

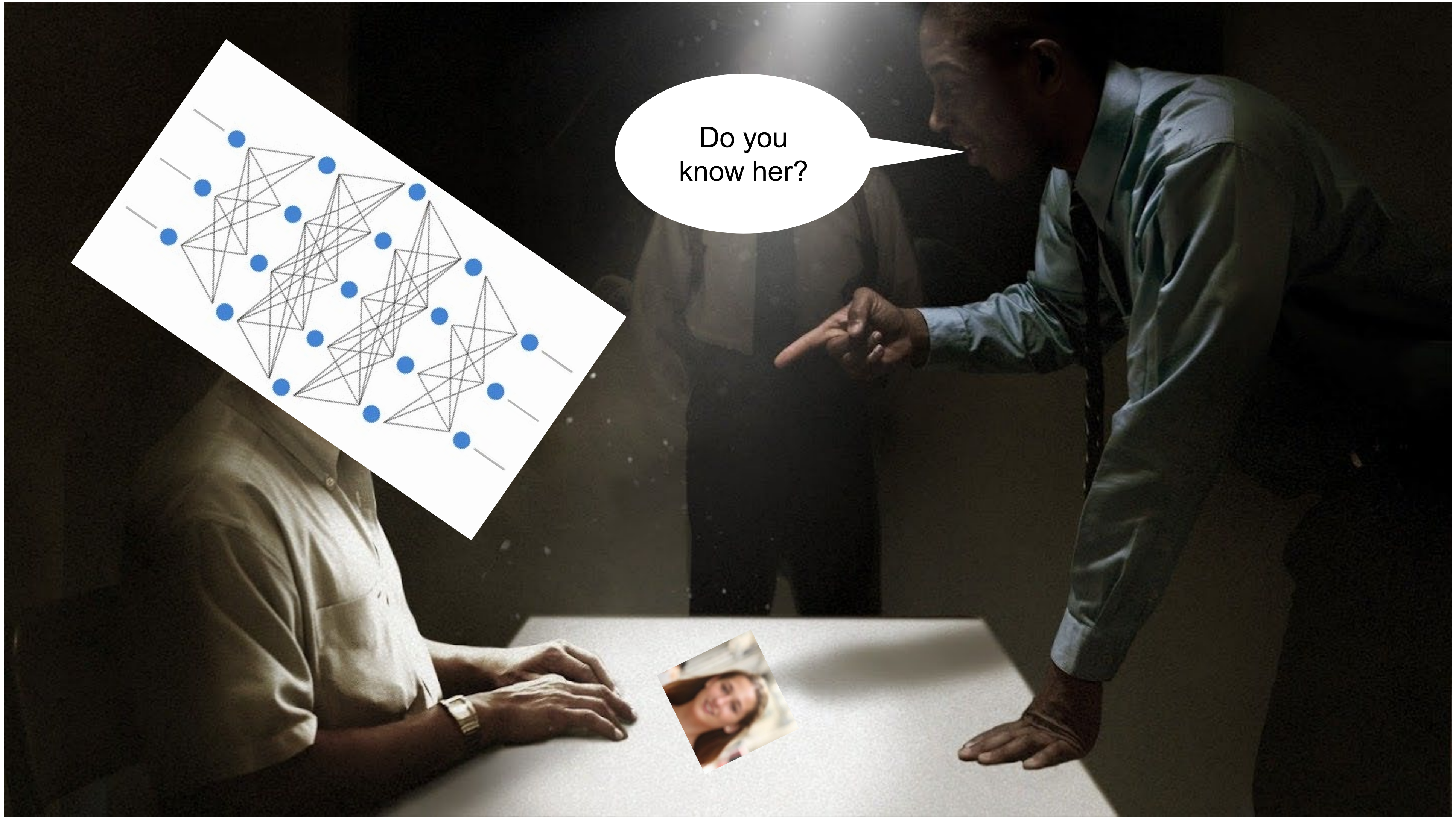
- **Attribute inference**: infer missing attributes of a partially known record used in the training dataset

- **Property inference**: infer statistics about the training set (e.g., percentage of women)

- **Gradient inversion**: (approximately) recover examples from the training set

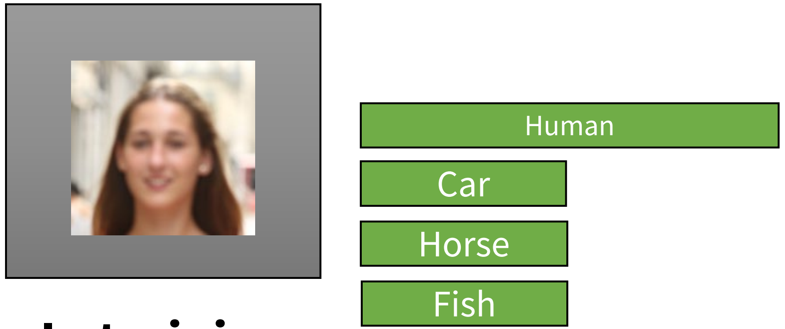
- **Model stealing:** recover the model (the weights)

Membership Inference attacks

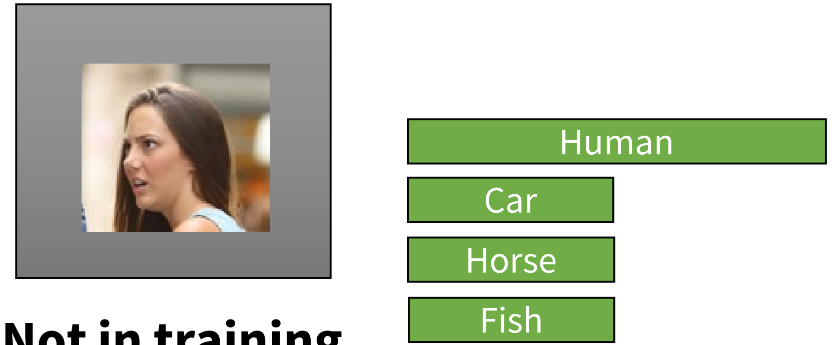


Membership inference – The intuition

Ideally, we would like: samples of **in** and **out** have the same confidence distribution

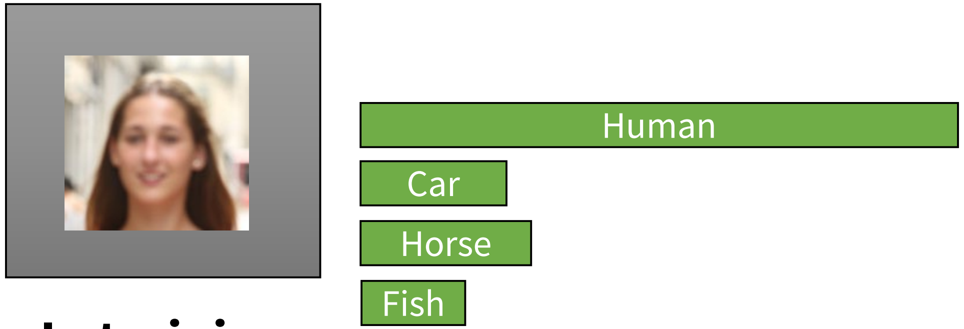


In training

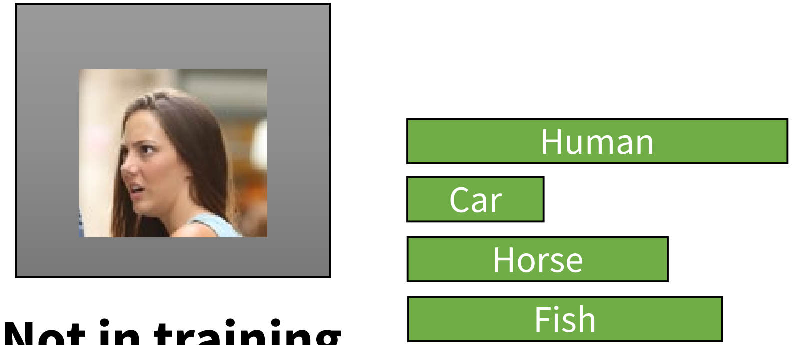


Not in training

However, the reality is:



In training



Not in training

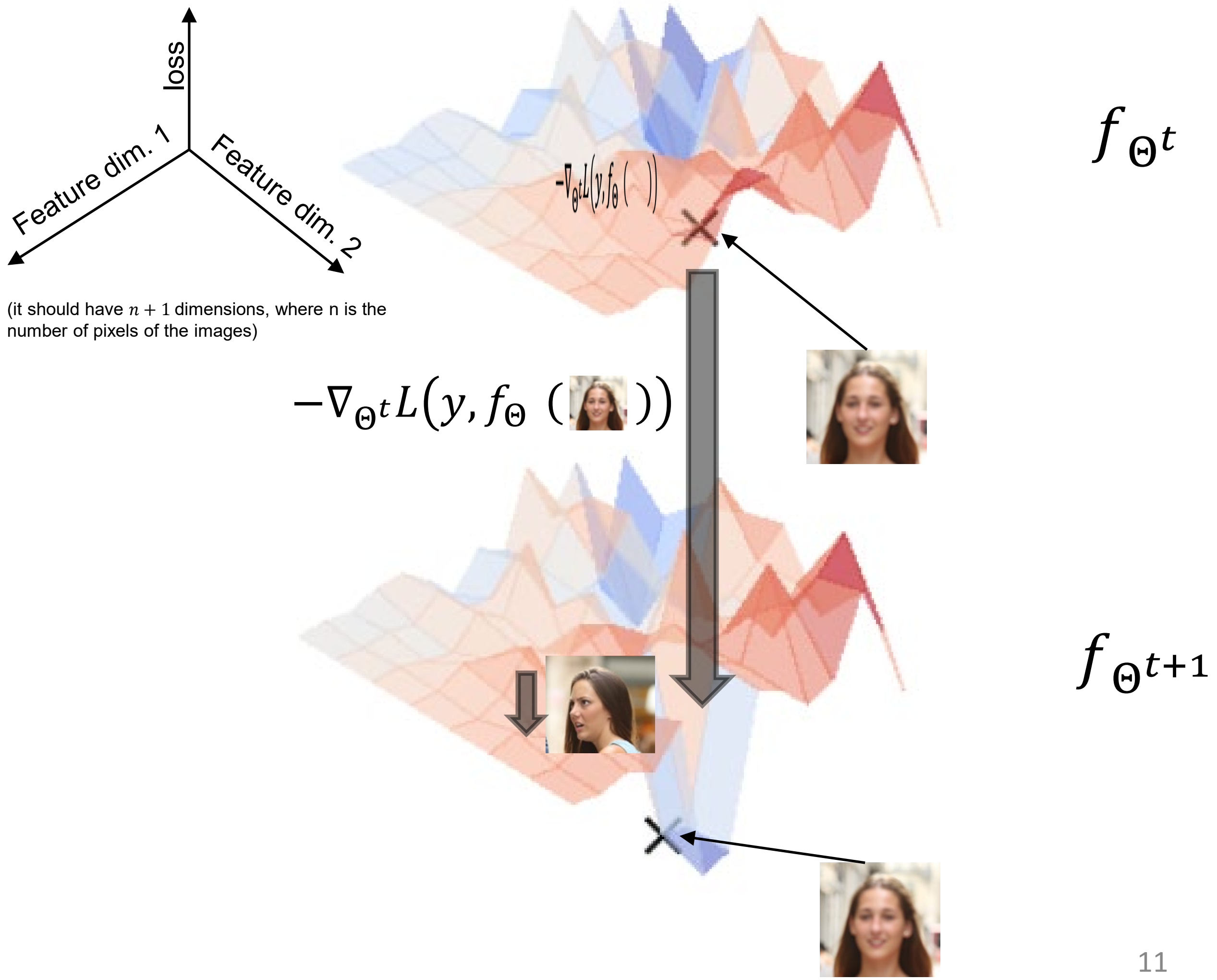
A byproduct of the training process:

Training process (SGD):

```
For $t$ in $[0,1,2, \dots , n]$:
1.  Sample batch from the training set:
        $x, y \sim X$
2.  Compute model's output (confidence score):
        $\tilde{y} = f_\Theta(x)$
3.  Compute loss:
        $l = L(y, \tilde{y})$
4.  Compute gradient:
        $\Delta = \nabla_{\Theta_t} l$
5.  Update parameters:
        $\Theta^{t+1} = \Theta - \eta\Delta$
```

X: training set
x, y: training instance and label
f: model
Θ: model's parameters
L: loss function
η: learning rate

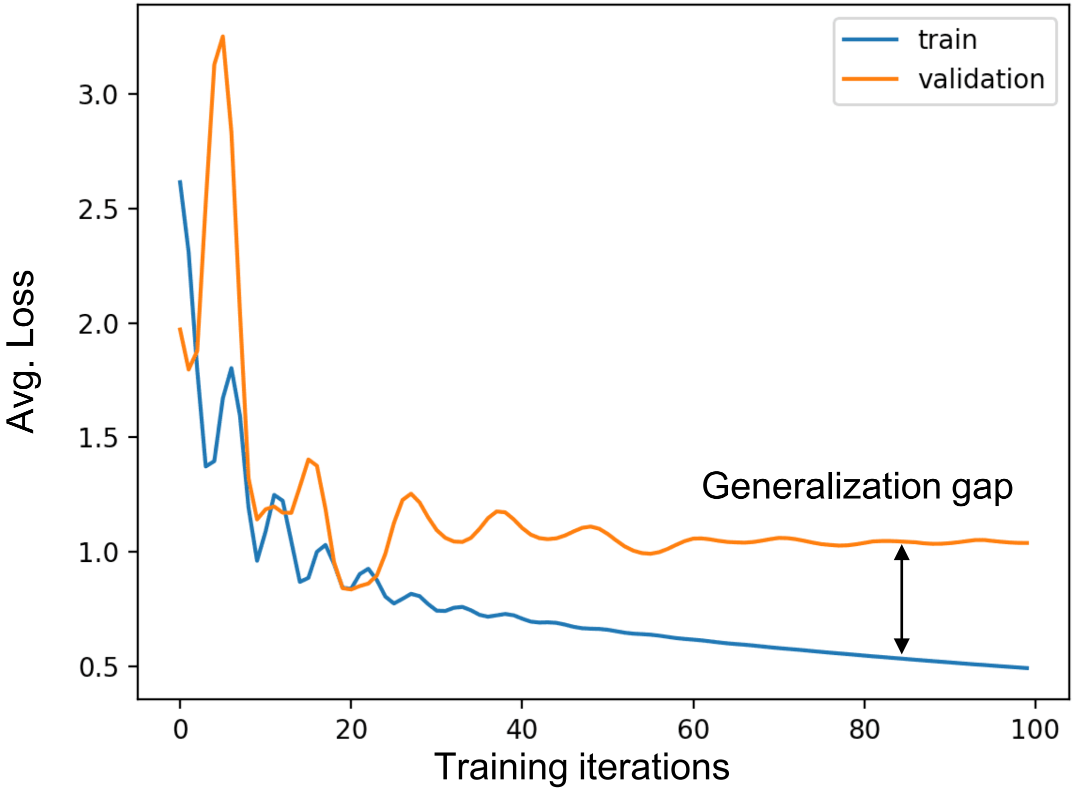
Loss surface:


loss
Feature dim. 1
Feature dim. 2
$-\nabla_{\Theta^t} L(y, f_\Theta( ))$
$-\nabla_{\Theta^t} L(y, f_\Theta( ))$
f_{Θ^t}
$f_{\Theta^{t+1}}$

(it should have $n + 1$ dimensions, where n is the number of pixels of the images)

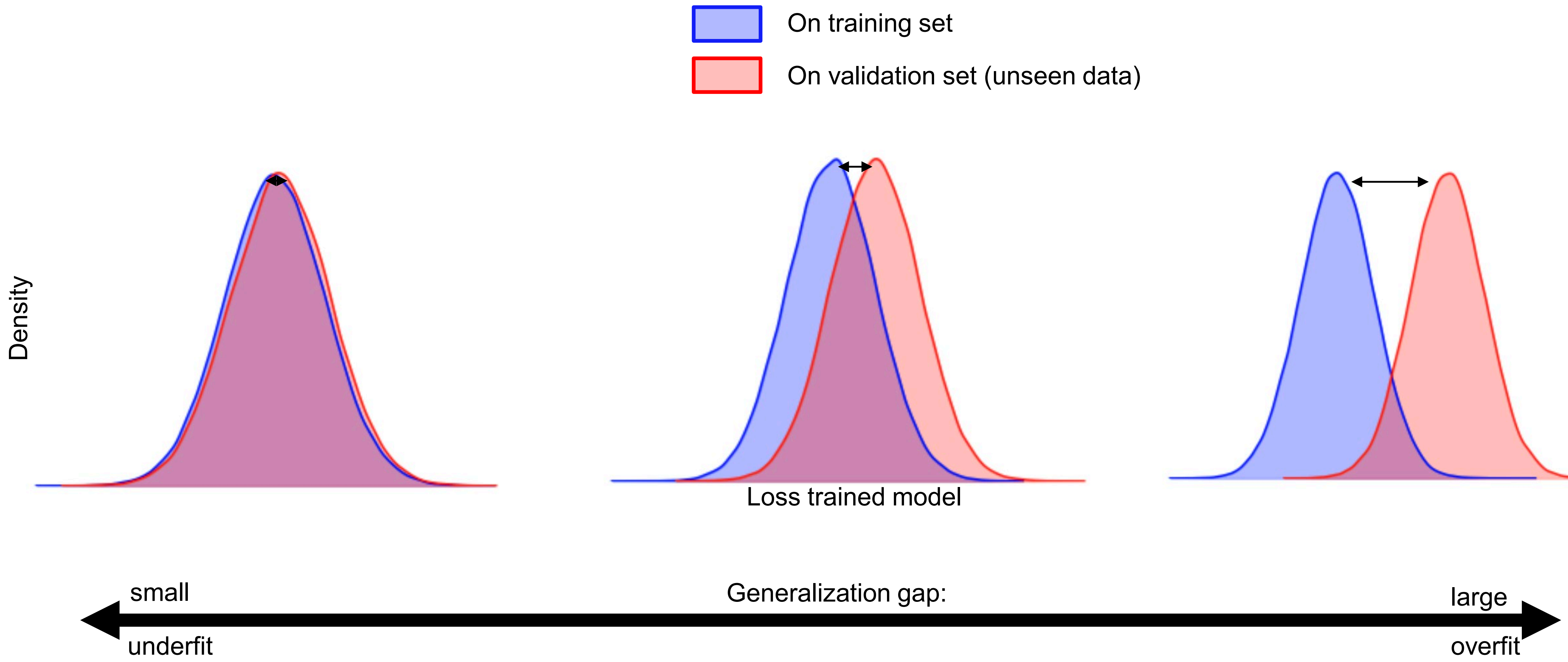
Generalization gap (overfitting) → privacy risk.

As long as there is generalization gap, there is privacy risk (a sufficient condition [Yeom CSF18]):

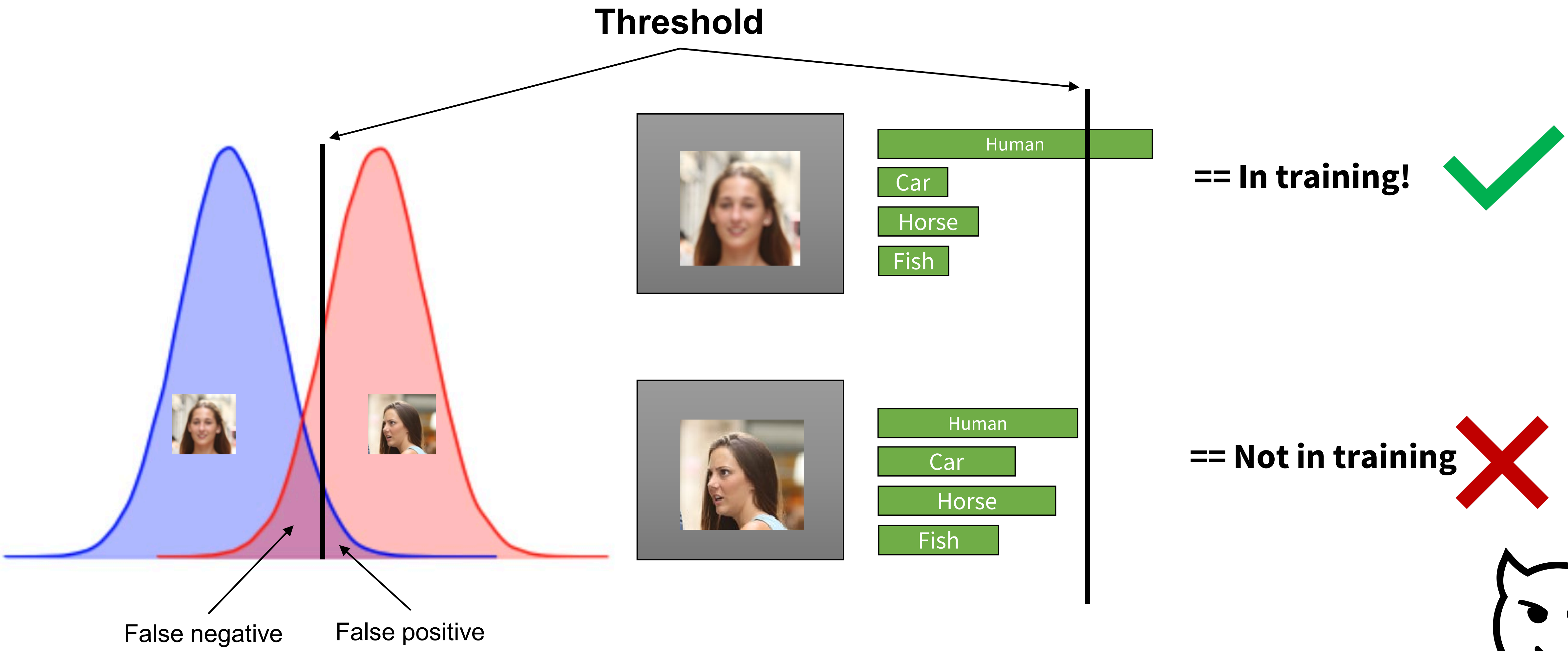


[Yeom CSF18] Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting

Generalization gap (overfitting) → privacy risk



If distributions are separable…



Threshold

Human

Car

Horse

Fish

== In training! ✓

Human

Car

Horse

Fish

== Not in training ✗

False negative

False positive

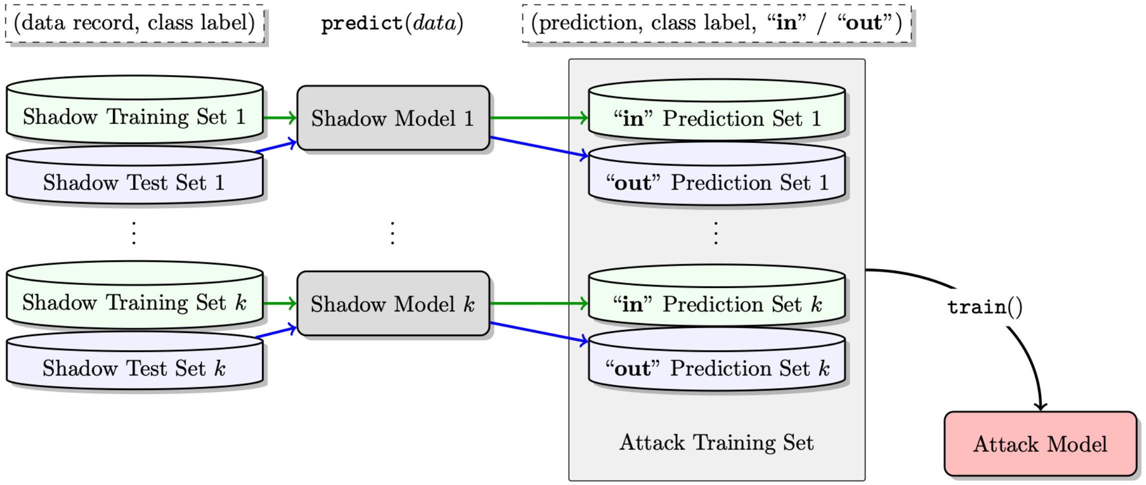
Why can we evaluate these?

We can estimate the training and validation distributions and learn to distinguish them

Example:

Long story short; The attacker:

1. Creates multiple models with the target's architecture.
2. Uses the shadow copies to create a supervised membership classifier.
3. Uses the classifier to infer membership of new points.



Membership inference – How bad is it? Quantifying leakage

- **Goal:** measuring the risk of a privacy breach
  - Can the adversary exploit the gap?

- **Questions**
  - What do we want to evaluate?
  - What should I measure?
  - How powerful is the adversary?

What do we want to evaluate?

- Evaluating the learning algorithm
  - How does membership inference work on the average dataset when using a learning algorithm – **Population attacks**

- Evaluating concrete leakage: model and target
  - How does membership inference work for a given model (a concrete output of the learning algorithm) and a given target (a concrete sensitive example in training) – **Reference attack**

What do we measure?

- What does it mean to succeed? When is there a privacy breach?

- Accuracy: how correct a model is (probability classifying a sample correctly)

What do we measure?

- What does it mean to succeed? When is there a privacy breach?
- Accuracy: how correct a model is (probability classifying a sample correctly)
- … but what if the adversary is very correct at identifying samples not in the model? High accuracy is not always a sign of a privacy breach

What do we measure?

- What does it mean to succeed? When is there a privacy breach?

- The adversary guesses correctly
  - True Positive Rate is high (probability that an actual positive will test positive)

- And the adversary does not guess incorrectly
  - False Positive Rate is low (probability an actual negative tests positive)

What do we measure?

- And for whom?

What do we measure?

- And for whom?

- Attacks have disparate impact:
  - Average examples tend to be protected – indistinguishable if one is in the dataset or not
  - Outliers tend to be vulnerable – there presence does skew the statistics!

- Looking at averages may not tell you the complete story

How powerful is the adversary?

- What does the adversary need to know to exploit the gap?
  - Worst case scenario:
    - access to a subset of the training set
    - knowledge of the architecture used
  - This are pretty strong assumption…

How powerful is the adversary?

- What does the adversary need to know to exploit the gap?
  - Worst case scenario:
    - access to a subset of the training set
    - knowledge of the architecture used
  - This are pretty strong assumption…

- Unfortunately signals are sufficiently strong and can be found
  - Even without data, and training on a different architecture…

How powerful is the adversary?

- What does the adversary need to know to exploit the gap?
  - Worst case scenario:
    - access to a subset of the training set
    - knowledge of the architecture used
  - This are pretty strong assumption…

- Unfortunately signals are sufficiently strong and can be found
  - Even without data, and training on a different architecture…

- **More importantly… you do not get to decide what the adversary knows/does**

Mitigations

Mitigations – Remove some data

- What is not in the model cannot be learned!
  - Still membership inference may be possible
- Negative effect on model performance

Mitigations – Remove outliers

- Average points are protected… let’s remove the vulnerable points

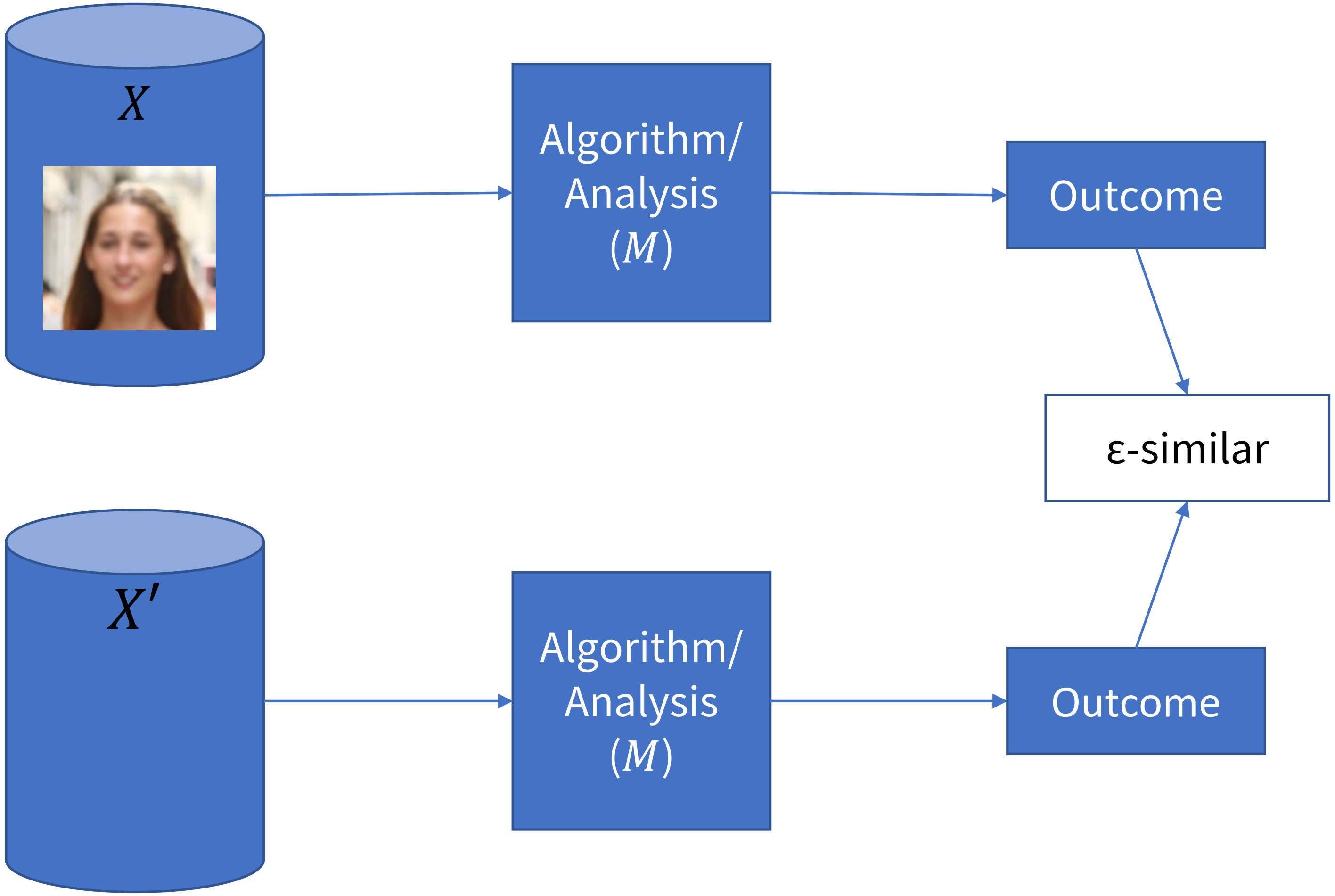
Mitigations – Remove outliers

- Average points are protected… let’s remove the vulnerable points
- but outliers depend on the distribution
  - When you remove an outlier, another point becomes an outlier!

Mitigations – Don’t gather the data

- Federated / decentralized learning
- Same attacks are possible!
  - And victims easier to isolate: less data -> easier attacks

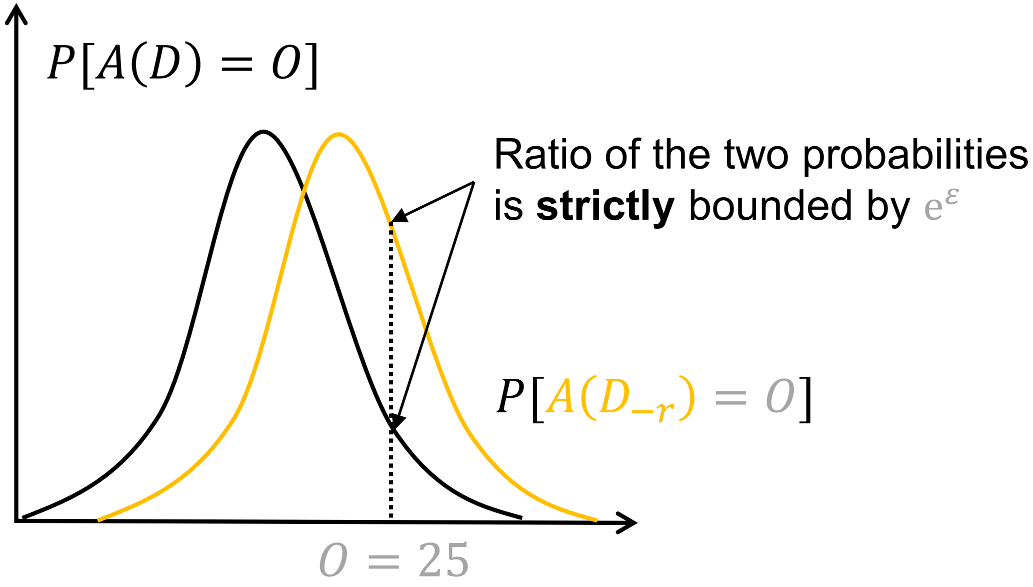
Differential Privacy



$$\frac{\mathrm{P}(\text{outcome}\,|X)}{\mathrm{P}(\text{outcome}\,|X')} \approx 1 + \varepsilon$$

$$\Pr[M(X) \in T] \leq e^{\varepsilon} \Pr[M(X') \in T]$$

Differentially Private Machine Learning


$P[A(D) = O]$
Ratio of the two probabilities is **strictly** bounded by e^{ε}
$P[A(D_{-r}) = O]$
$O = 25$

For any neighbouring databases D, D_{-r} and any possible output O

$$\log \frac{P[A(D) = O]}{P[A(D_{-r}) = O]} < \varepsilon$$

Differential privacy in Machine Learning

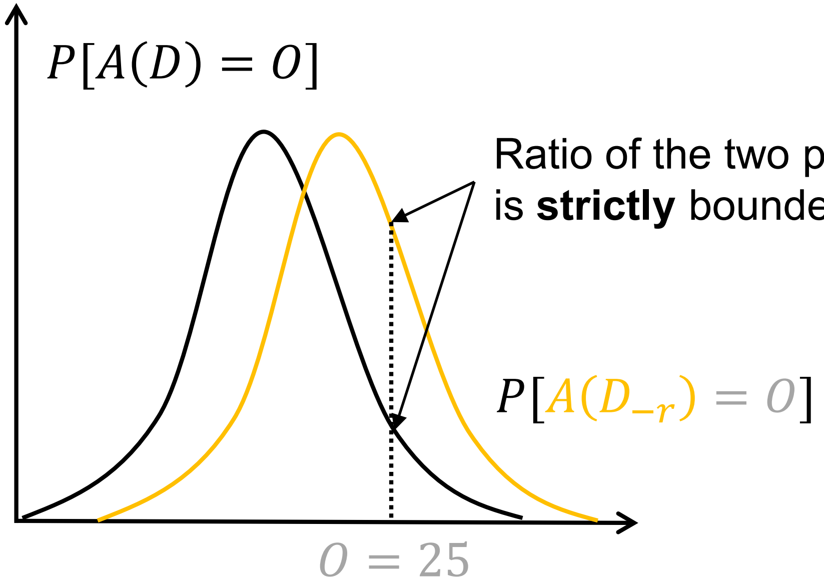
- Three ways to apply Differential Privacy to Machine Learning
  - Objective Perturbation
  - Gradient Perturbation
  - Output Perturbation

- For complex learning tasks (deep learning), we can not derive sensitivity bounds for the objective and output and have to use gradient perturbation

Differentially Private Machine Learning

Composition

Naïve Composition: $\boldsymbol{\varepsilon - DP}$



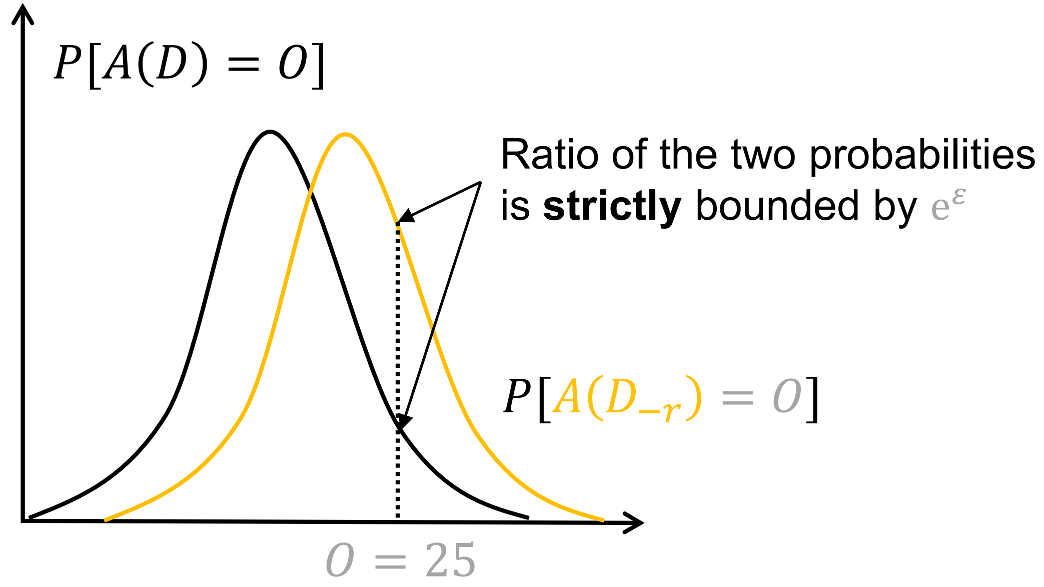
Ratio of the two probabilities is **strictly** bounded by e^{ε}

For any neighbouring databases D, D_{-r} and any possible output O

$$\log \frac{P[A(D) = O]}{P[A(D_{-r}) = O]} < \varepsilon$$

Differentially Private Machine Learning

Composition



$P[A(D) = O]$
Ratio of the two probabilities is **strictly** bounded by e^{ε}
$P[A(D_{-r}) = O]$
$O = 25$

For any neighbouring databases D, D_{-r} and any possible output O

$$\log \frac{P[A(D) = O]}{P[A(D_{-r}) = O]} < \varepsilon$$

Naïve Composition: $\boldsymbol{\varepsilon - DP}$

Privacy decreases linearly as we perform differentially private operations

$$\varepsilon = \varepsilon_1 + \cdots + \varepsilon_N$$

Advanced Composition: $\boldsymbol{(\varepsilon, \delta) - DP}$

For any neighbouring databases D, D_{-r} and any possible output O

$\log \frac{P[A(D)=O]}{P[A(D_{-r})=O]} < \varepsilon$ **with probability $\boldsymbol{1 - \delta}$**

Differential Privacy trade-offs

Fundamental issue:

- By design, DP mechanisms partially destroy information

The promise:

- As long as there is enough data is big enough, the noise introduced by the DP mechanism "cancels out“. Only trends that are relevant to many people are visible.

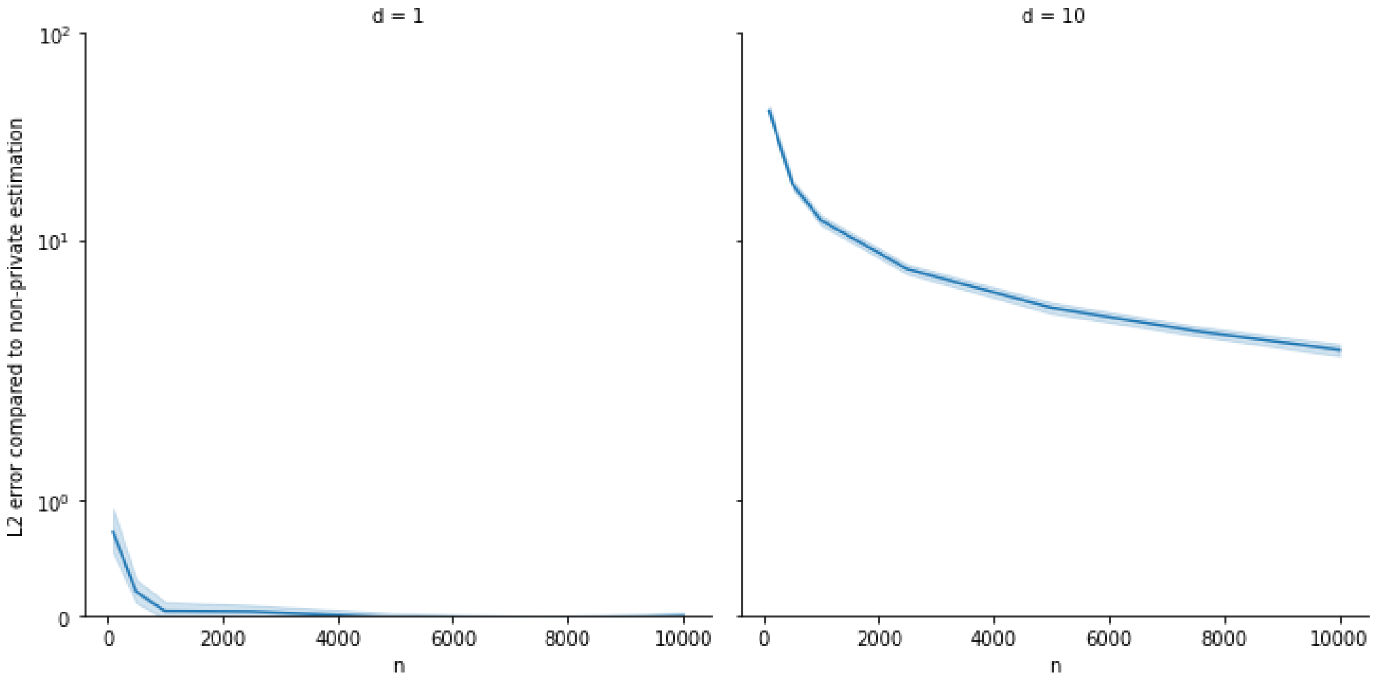
What can go wrong?

Example: local differential privacy

Application: ~~Machine Learning~~ Mean estimation of binary-valued vectors

LDP mechanism: each data contributor adds a Laplace random noise

- Sensitivity = d, Epsilon = 0.1 -- zero-centered noise => eventually it cancels out



At around 500 examples, private mean has similar error to the non-private mean.

10000 examples and private mean still has much higher error as the non-private mean.

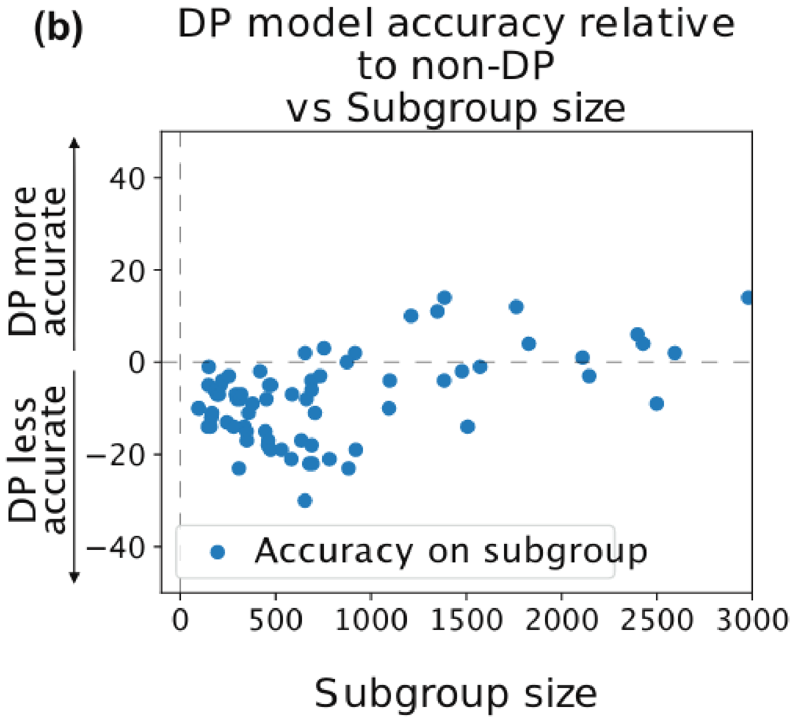
Differential Privacy needs big data

- You need a lot more data than usual to compensate for the noise added in a DP mechanism
- If you work with small datasets, DP is likely to destroy utility
  - What is small? Depends on the dimensionality
  - E.g., in the local model, LDP reduces effective sample size for convex optimization from *n* to $\varepsilon^2 n / d$.
    "Local Privacy and Statistical Minimax Rates" Duchi et al. 2013
- In this case, other approaches to privacy protections should be used, e.g., not using a learning model at all.

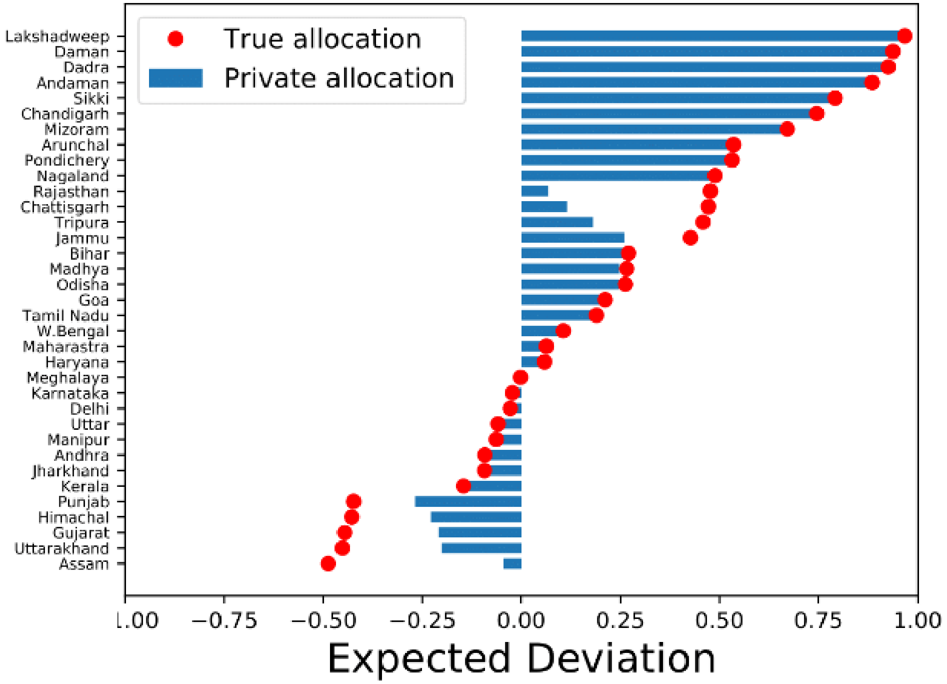
- What can still go wrong if you *do* have a lot of data?

Differential Privacy’s disparate impact

The impact of DP on accuracy may differ across subgroups of the population



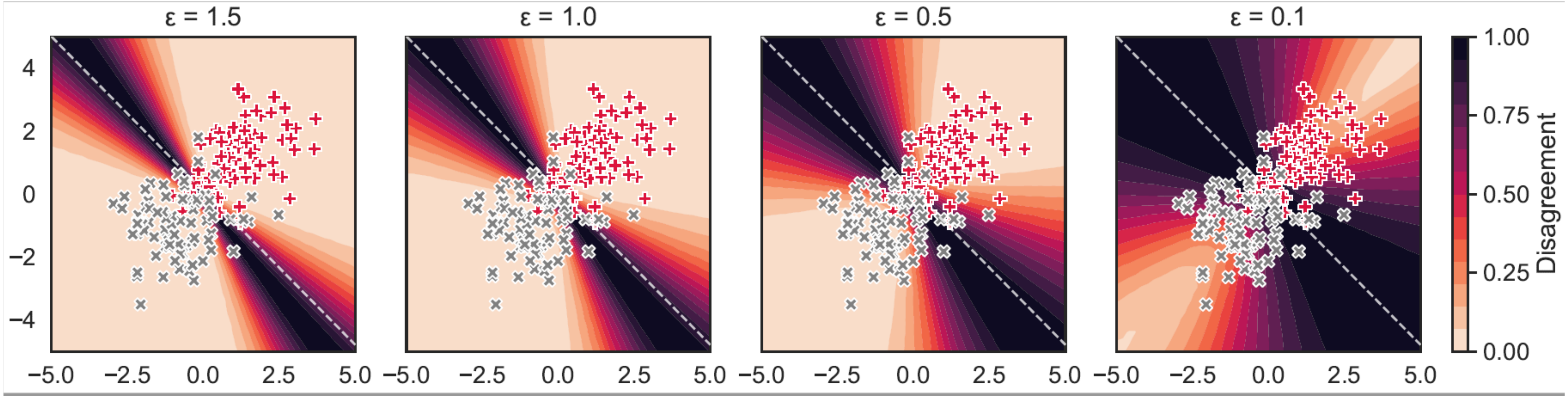
Disparate impact of DP on a computer vision problem trained with DP-SGD, epsilon ≈ 6
"Differential Privacy Has Disparate Impact on Model Accuracy"
Eugene Bagdasaryan, Vitaly Shmatikov 2019



Disparate impact of hypothetical Indian parliament seat apportionment if Census data had central Laplace "Fair Decision Making Using Privacy-Protected Data"
David Pujol et al. 2020

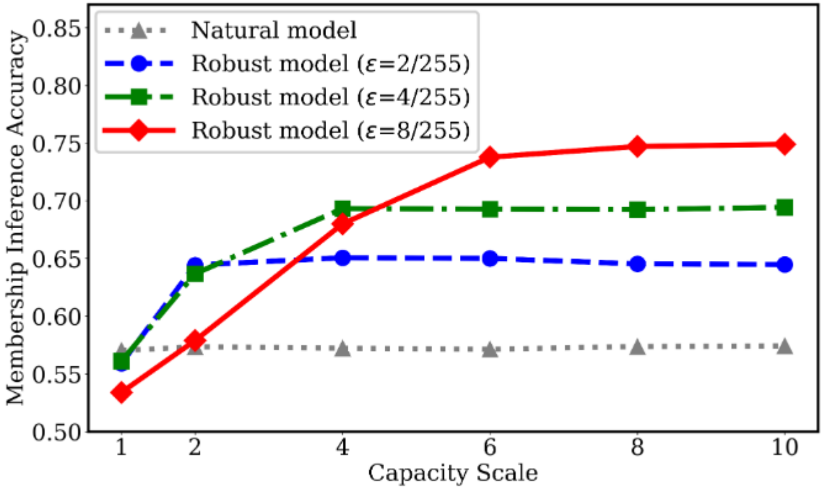
Differential Privacy increases arbitrariness

Models with same privacy level can give different results for a given example

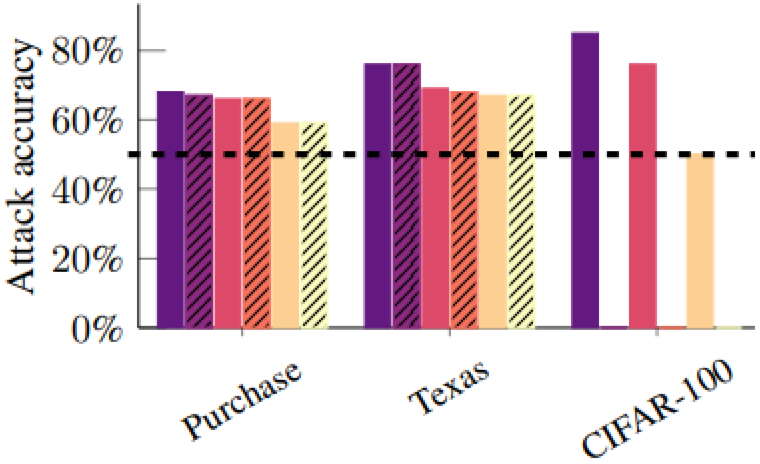


Privacy vs. everything else

Different *trustworthy* properties of ML and privacy interact in non-trivial ways



Adversarial training can make membership inference easier
"Privacy Risks of Securing Machine Learning Models against Adversarial Examples"
Liwei Song, Reza Shokri, Prateek Mittal 2019



Same for local explanation techniques
"On the Privacy Risks of Model Explanations"
Reza Shokri, Martin Strobel, Yair Zick 2019

Take aways

- **A ML model is the data used to train it**.
- Thus, if the training data is sensitive, the model is sensitive as well.

- If the training data is sensitive, one **must** take actions to limit the leakage.
- If no formal protections are used (e.g., DP), sharing the trained model (or gradient) can be as leaky as sharing the raw data.
- Yet, formal protections come with heavy cost in performance, utility and bias.