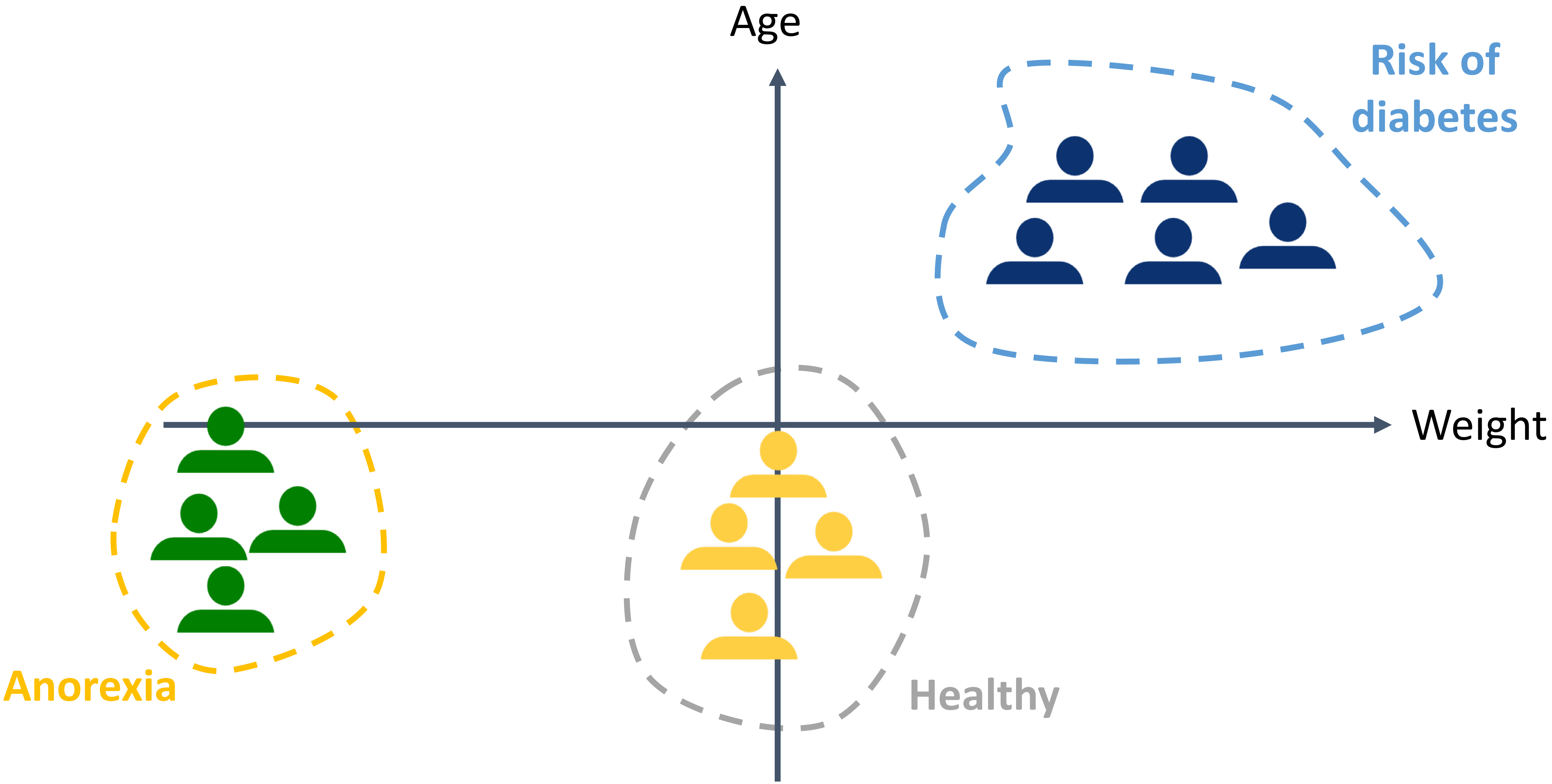
A Marauder's Map of Security and Privacy in Machine Learning

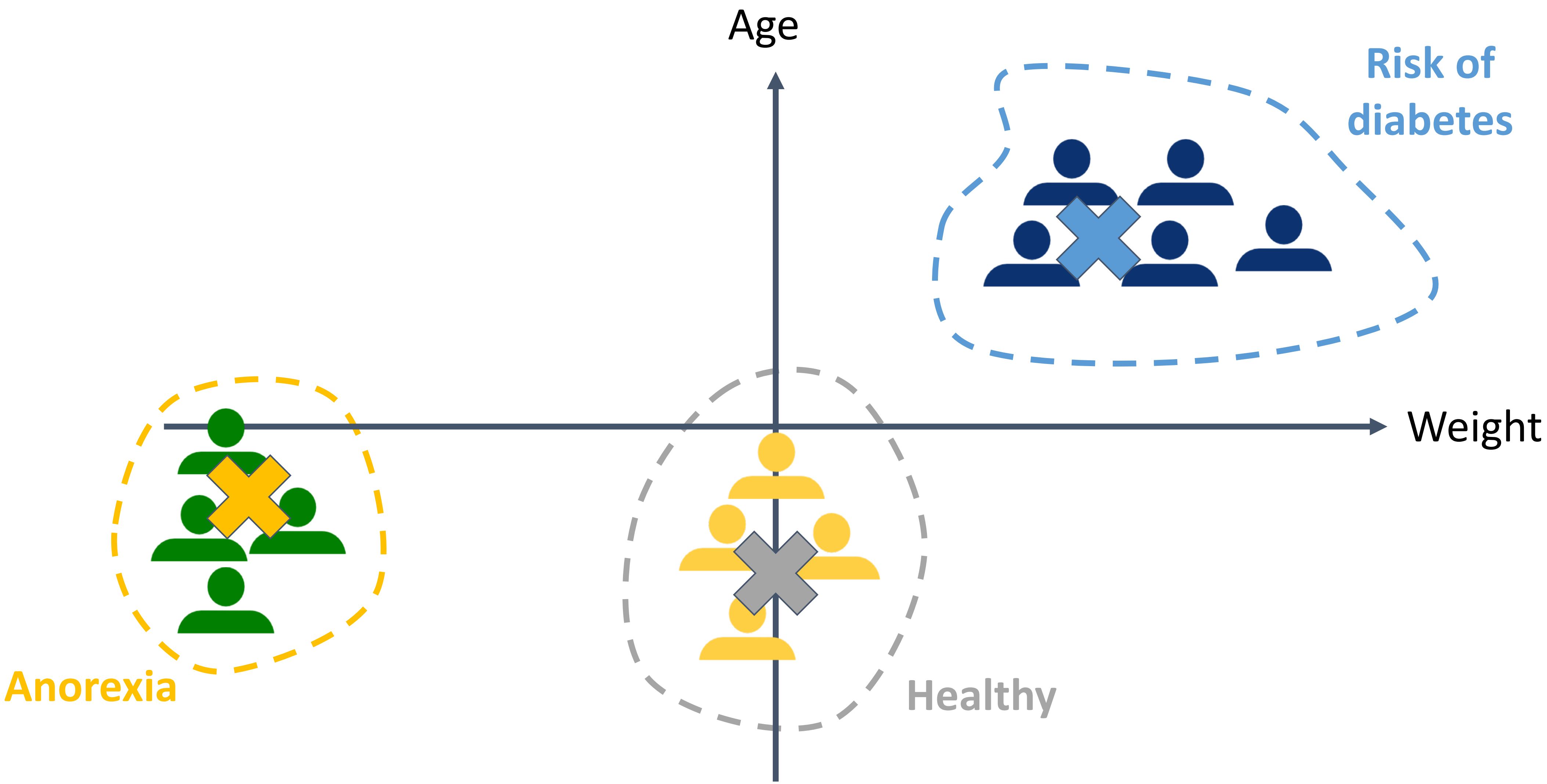
Nicolas Papernot

University of Toronto & Vector Institute

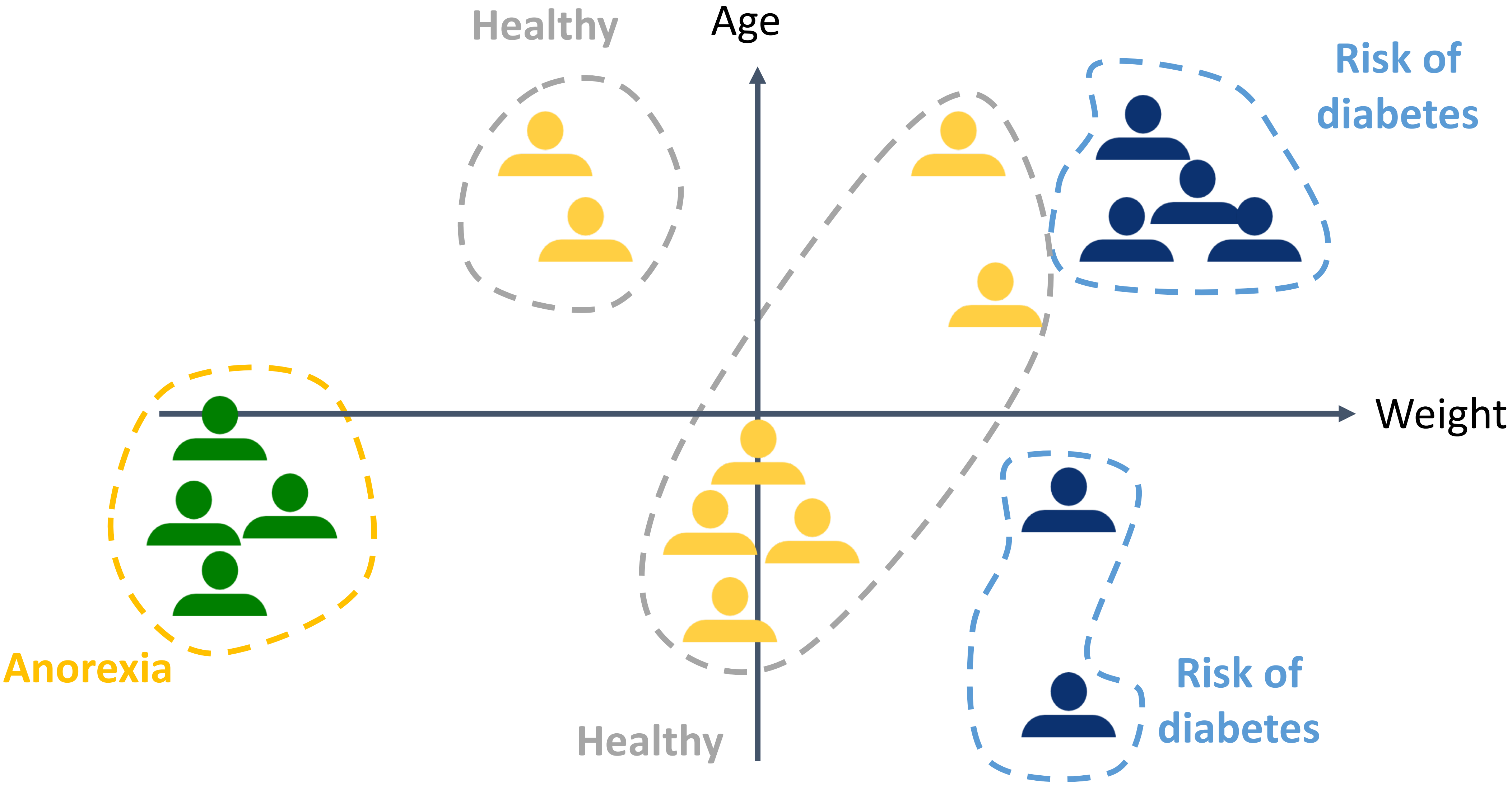
Machine learning is not magic: *ideal setting*



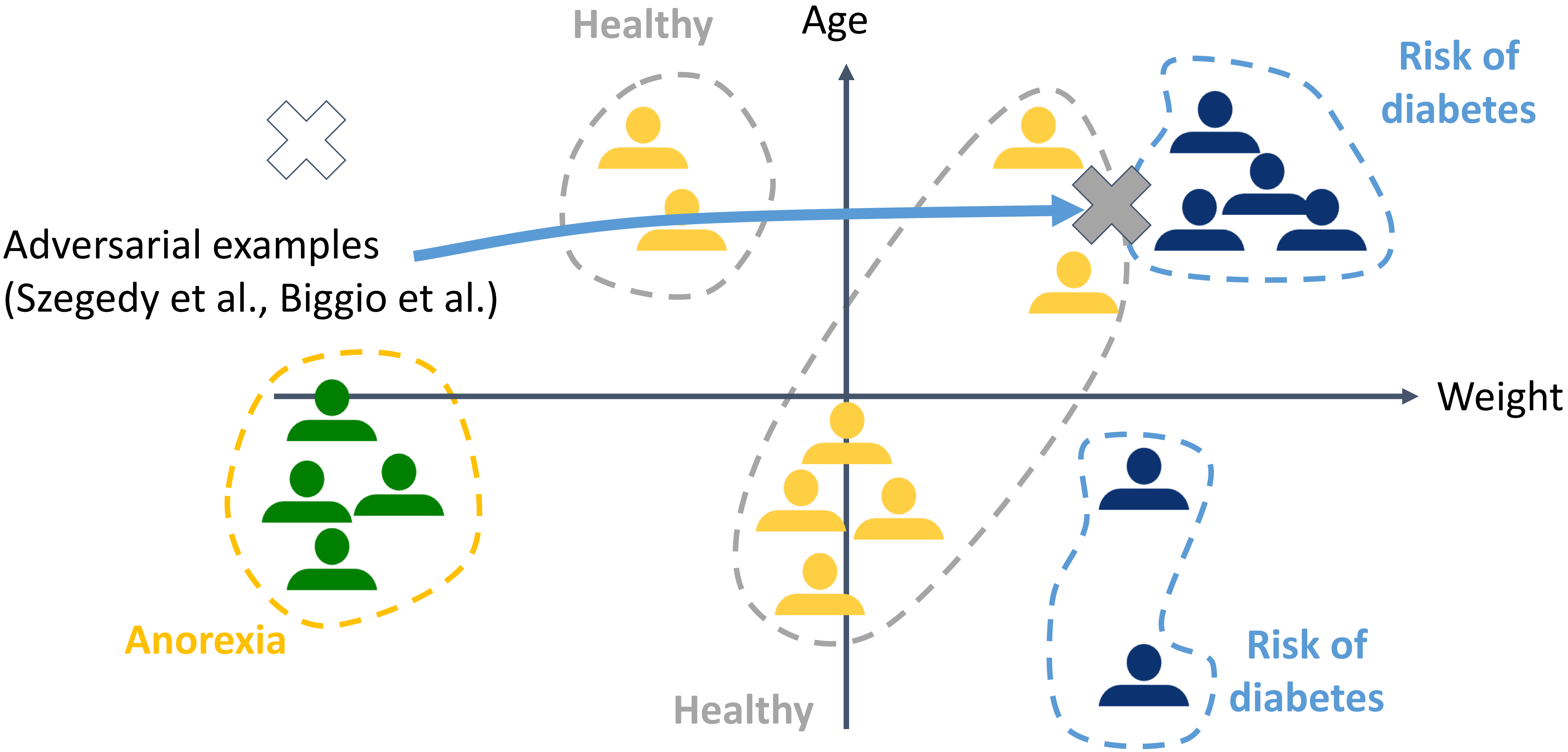
Machine learning is not magic: *ideal setting*



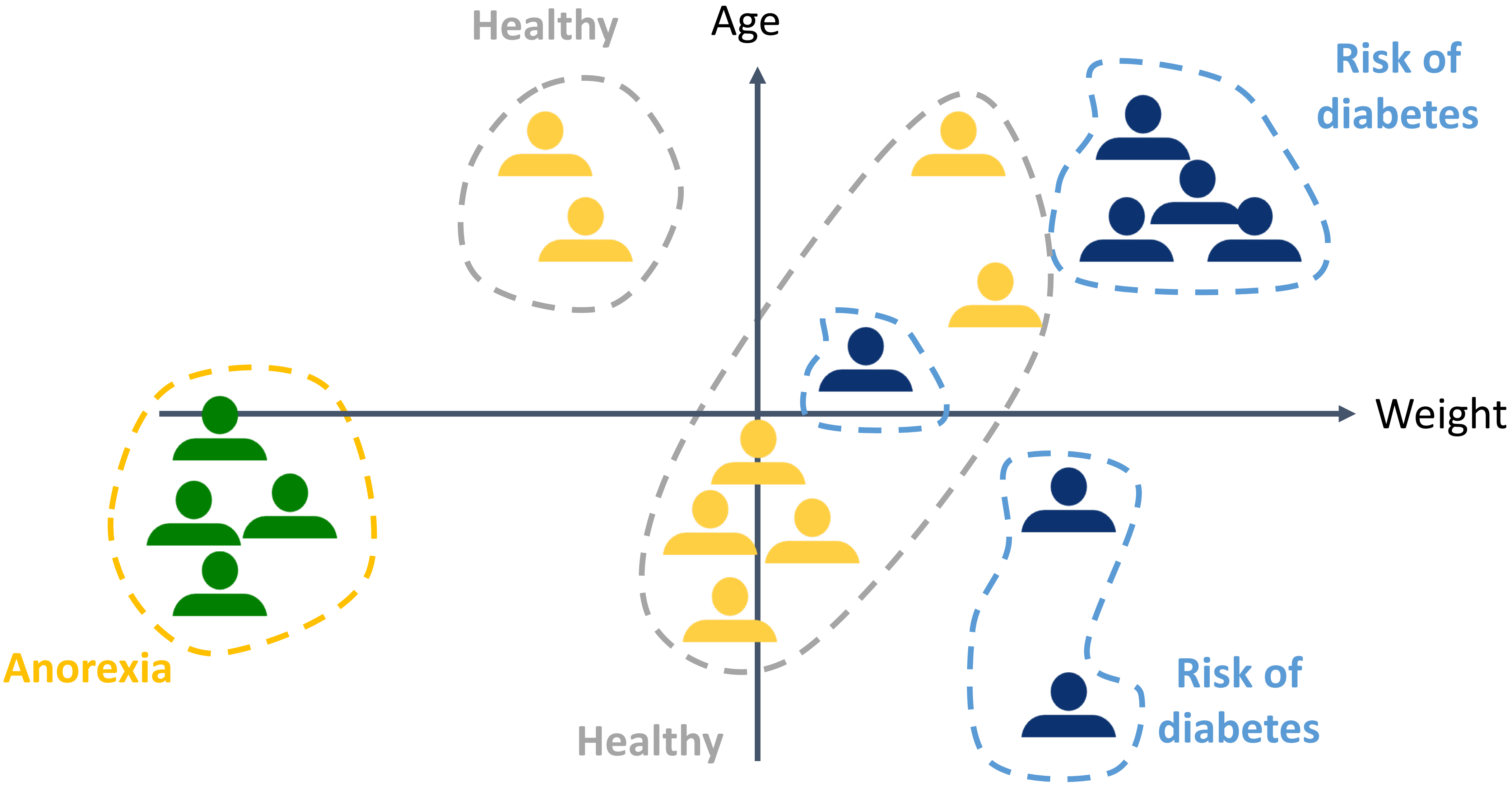
Machine learning is not magic: *(adversarial) real-world*



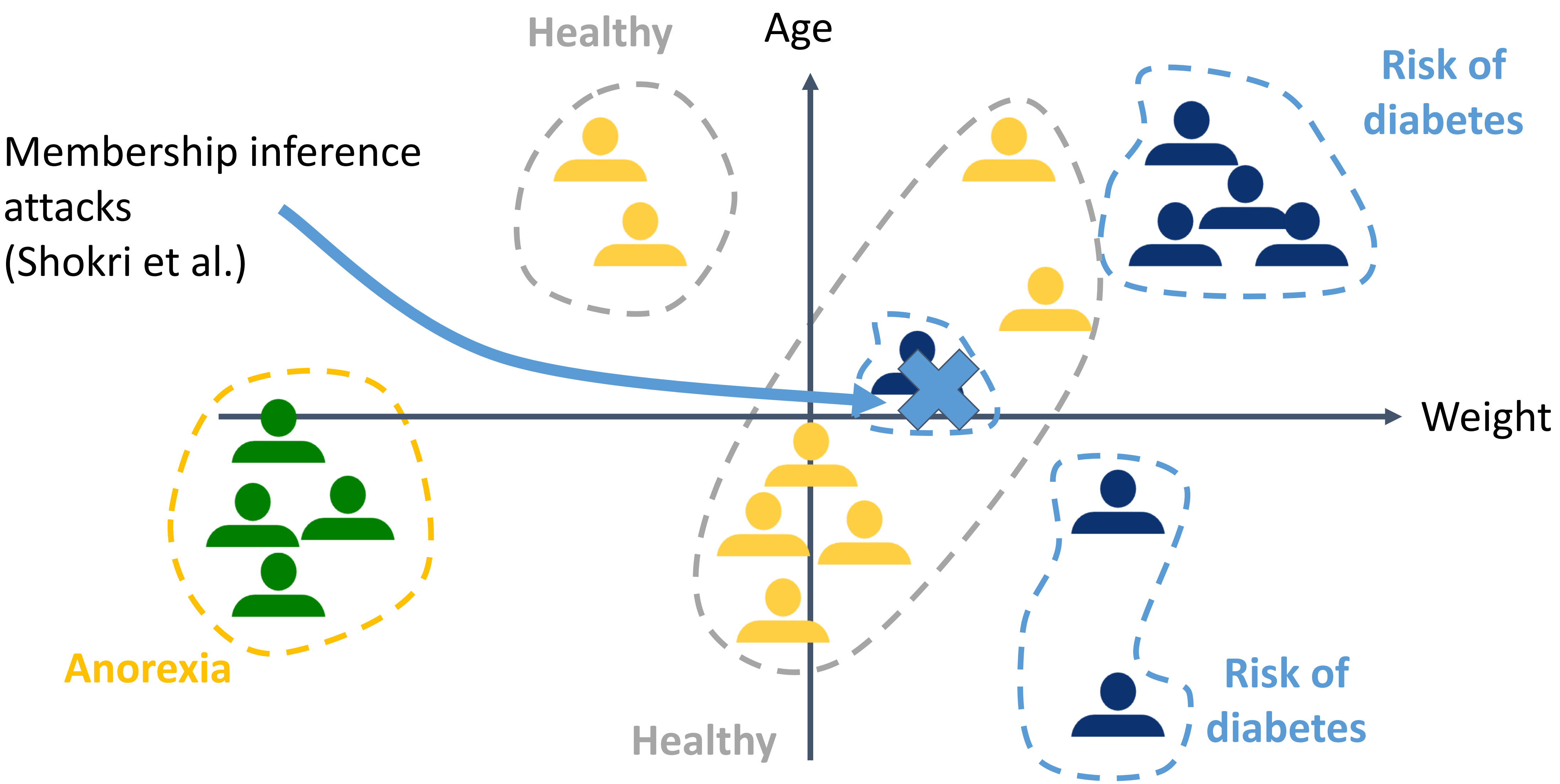
Machine learning is not magic: *(adversarial) real-world*



Machine learning is not magic: *(adversarial) real-world*



Machine learning is not magic: *(adversarial) real-world*



The ML paradigm in adversarial settings

Training

Data privacy attacks

Model parameters

Data poisoning

Training data

“Healthy”

Predictions

Model theft

Test data

Adversarial examples

Inference

Adapted from a slide by Ian Goodfellow

Is ML security any different from ~~real-world~~ computer security?

*"Practical security balances the cost of protection and the risk of loss, which is the cost of recovering from a loss times its probability" (*Butler Lampson, 2004)

Is the ML paradigm fundamentally different in a way that enables systematic approaches to security and privacy?

Revisiting Saltzer and Schroeder's principles

Saltzer and Schroeder's principles

Economy of mechanism.
Keep the design of security mechanisms simple.

Fail-safe defaults.
Base access decisions on permission rather than exclusion.

Complete mediation.
Every access to an object is checked for authority.

Open design.
The design of security mechanisms should not be secret.

Separation of privilege.
A protection mechanism that requires two keys to unlock is more robust and flexible.

Least privilege.
Every user operates with least privileges necessary.

Least common mechanism.
Minimize mechanisms depended on by all users.

Psychological acceptability.
Human interface designed for ease of use.

Work factor.
Balance cost of circumventing the mechanism with known attacker resources.

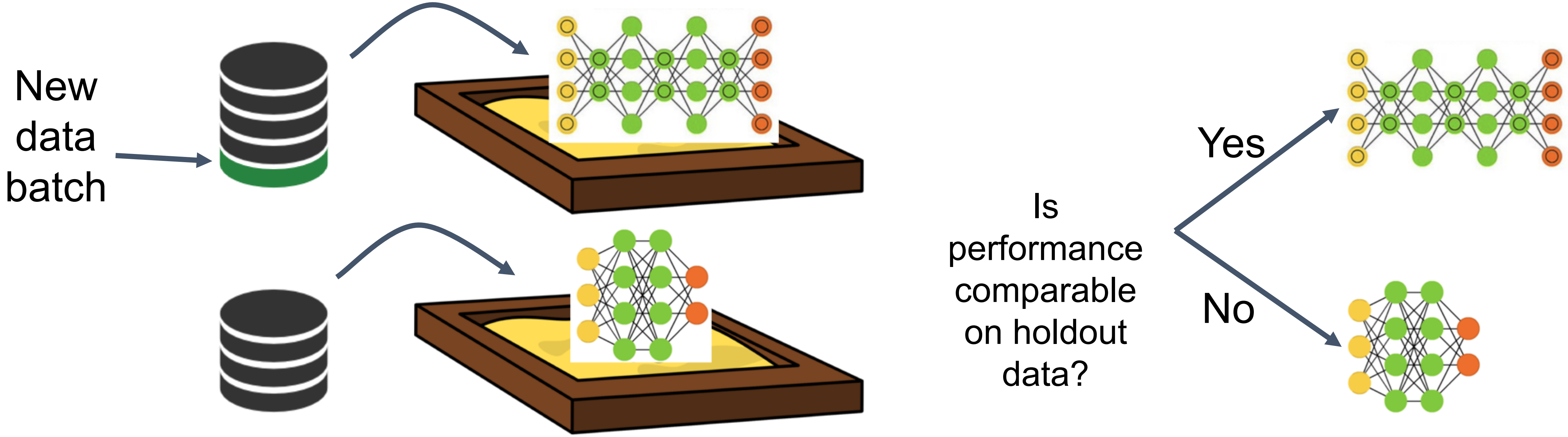
Compromise recording.
Mechanisms that reliably record compromises can be used in place of mechanisms that prevent loss.

Fail-safe defaults

Example 1: do not output low-confidence predictions at test time

Example 2: mitigate data poisoning resulting in a distribution drift

Attacker: submits poisoned points to gradually change a model's decision boundary
Defender: compares accuracy on holdout validation set ***before*** applying gradients

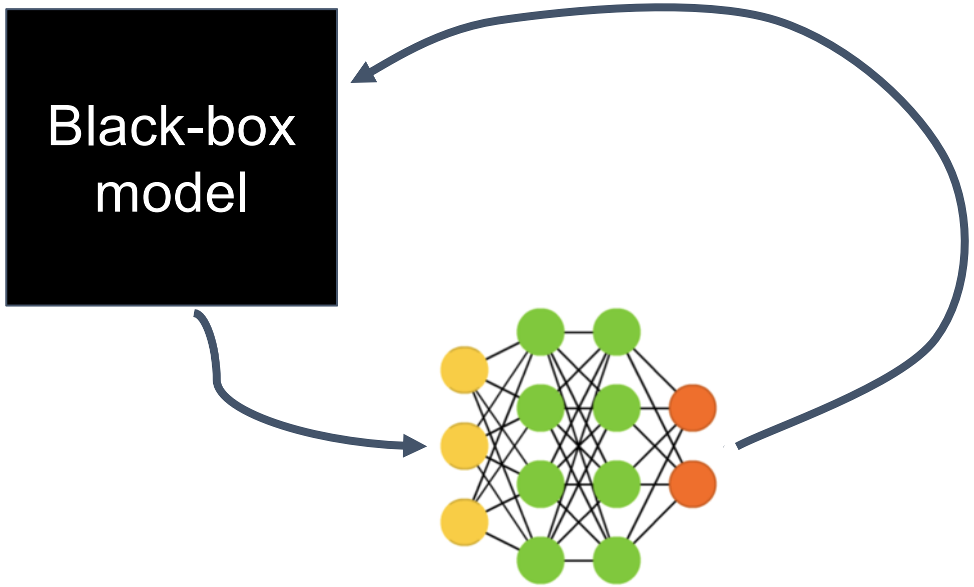


Open design

Example 1: black-box attacks are not particularly more difficult than white-box attacks

Insider leaks model

Reverse engineering



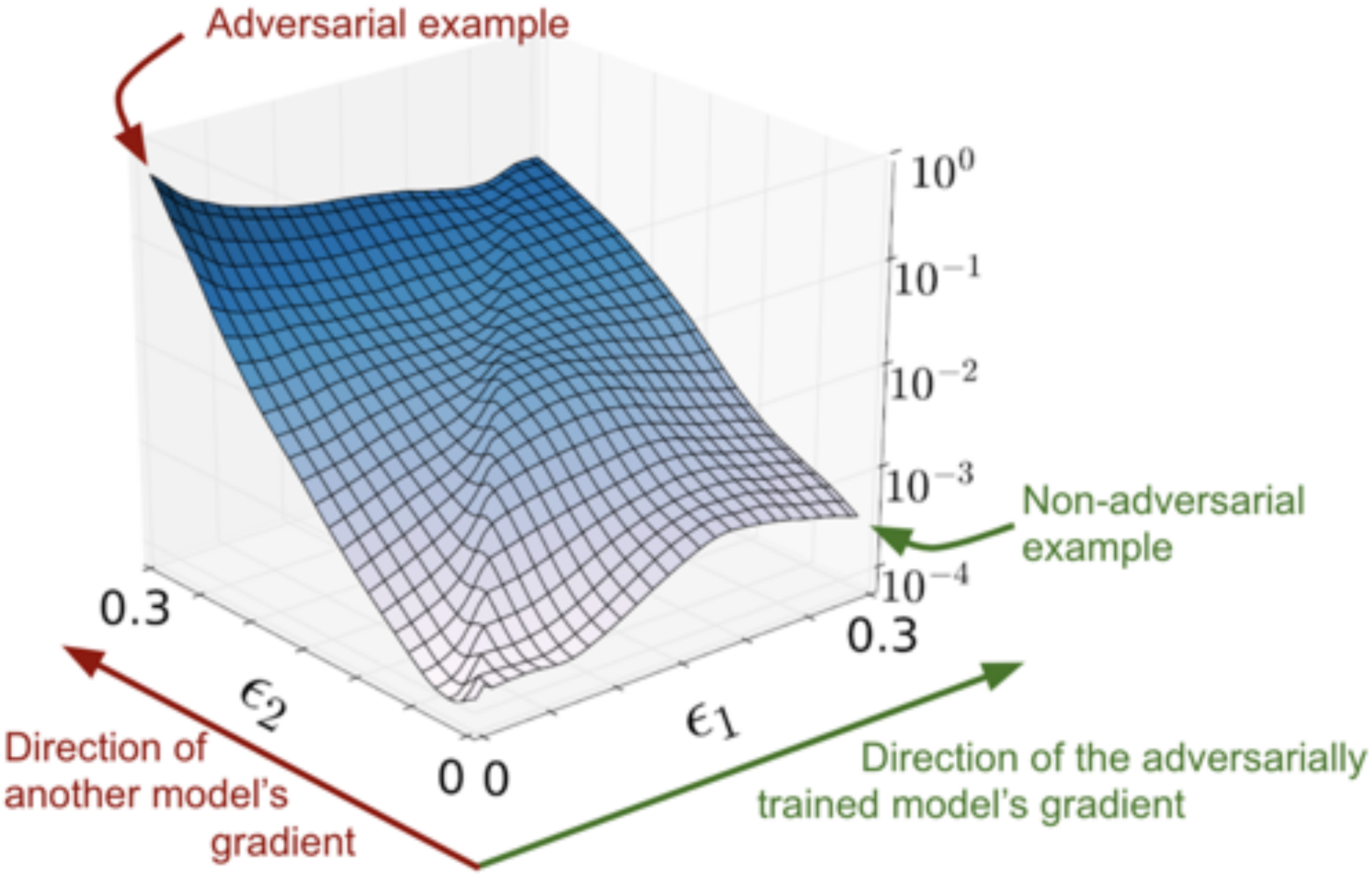
Model extraction

Transferability

ACM:2650798 (Šrndic and Laskov); arXiv:1602.02697 (Papernot et al.)

Open design

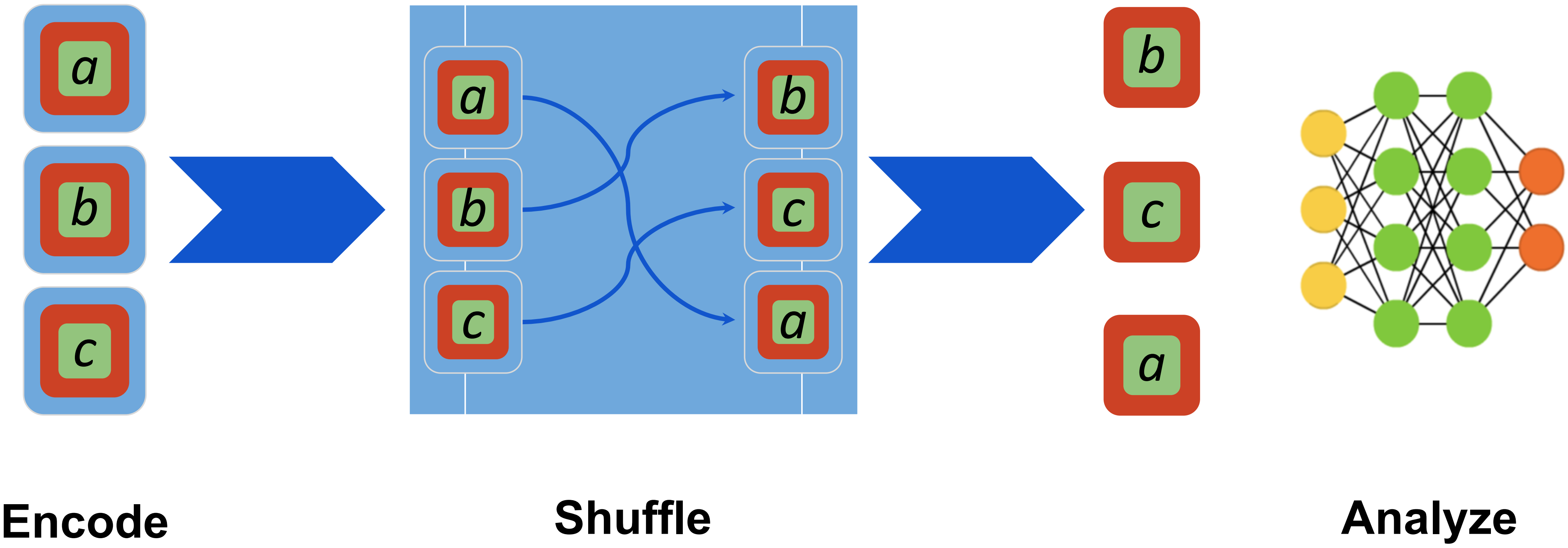
Example 2: gradient masking can be circumvented by a black-box attack



arXiv:1602.02697 (Papernot et al.); arXiv:1705.07204 (Tramer et al.); arXiv:1802.00420 (Athalye et al.)

Separation of privilege

Privacy can be obtained in the **data pipeline** through federated learning or by having different parties encode, shuffle and analyze data in ESA.



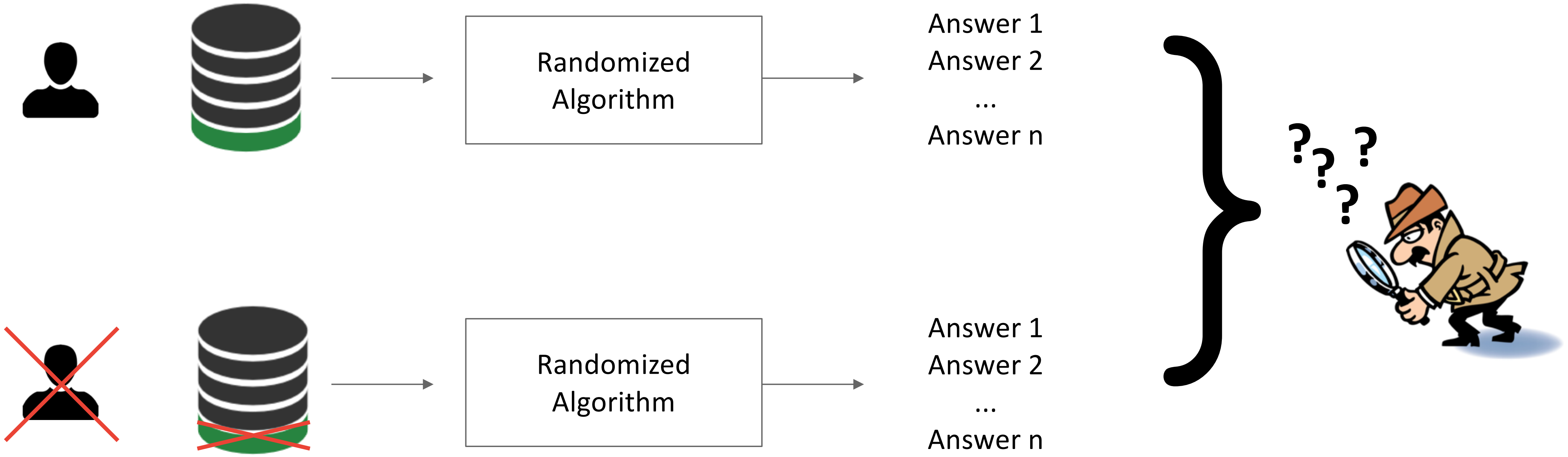
arXiv:1710.00901 (Bittau et al.); arXiv:1602.05629 (McMahan et al.)

Psychological Acceptability and Privacy in Machine Learning

What is a private algorithm?

Designing algorithms with privacy guarantees understood by humans is difficult.

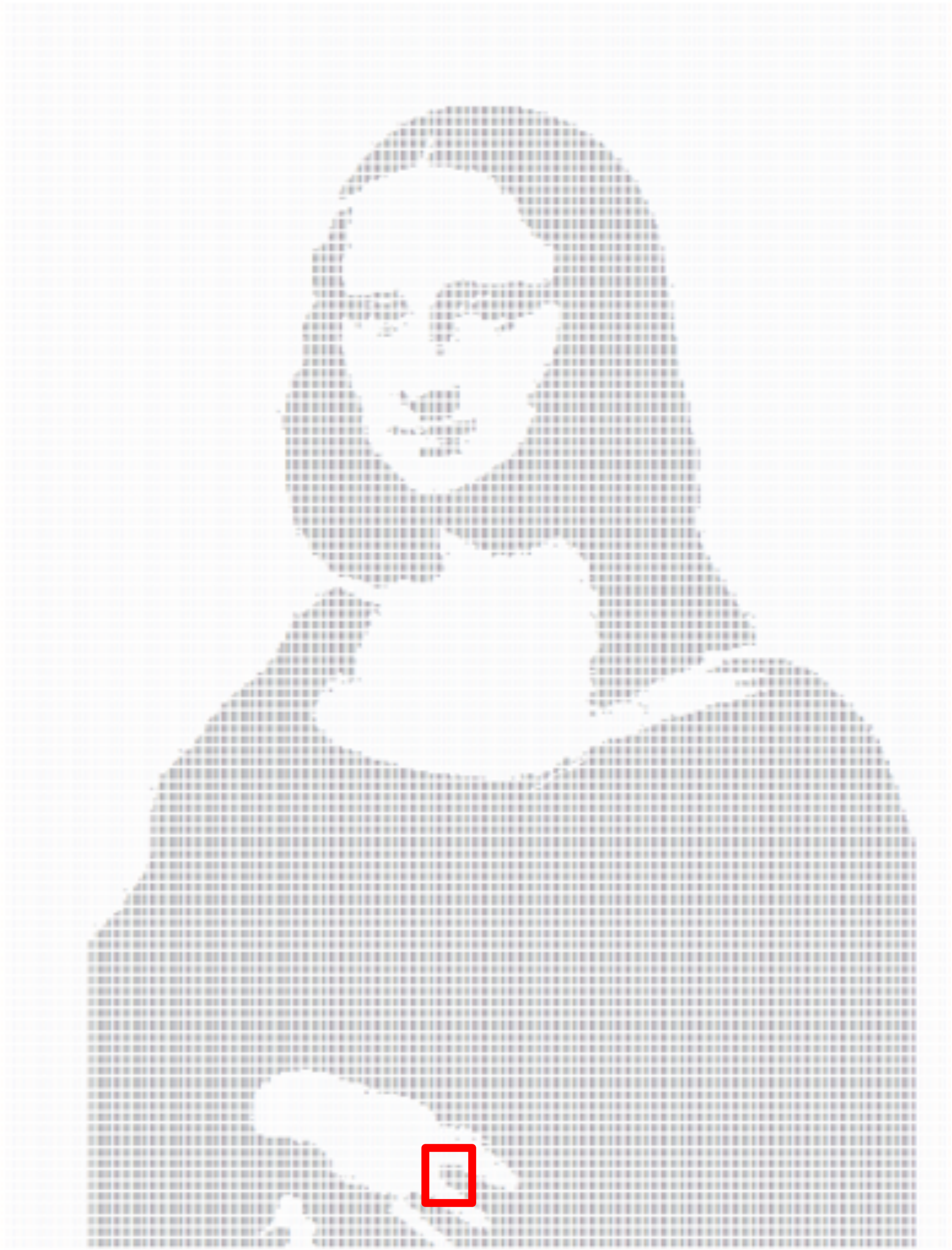
First question: how should we define privacy? Gold standard is now differential privacy.



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

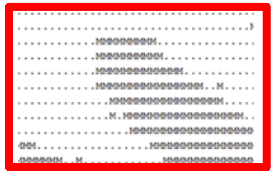
IACR:3650 (Dwork et al.)

A Metaphor For Private Learning



Slides adapted from Ulfar Erlingsson

An Individual's Training Data



```
....................................
...................................M
............MMMMMMMMM...............
............MMMMMMMMMM..............
............MMMMMMMMMMMMM...........
............MMMMMMMMMMMMMMMM..M.....
..............MMMMMMMMMMMMMMMMM.....
..............M.MMMMMMMMMMMMMMMMMM..
.................MMMMMMMMMMMMMMMMMMM
MMM.................MMMMMMMMMMMMMMMM
MMMMMMM..M...........MMMMMMMMMMMMMMM
```

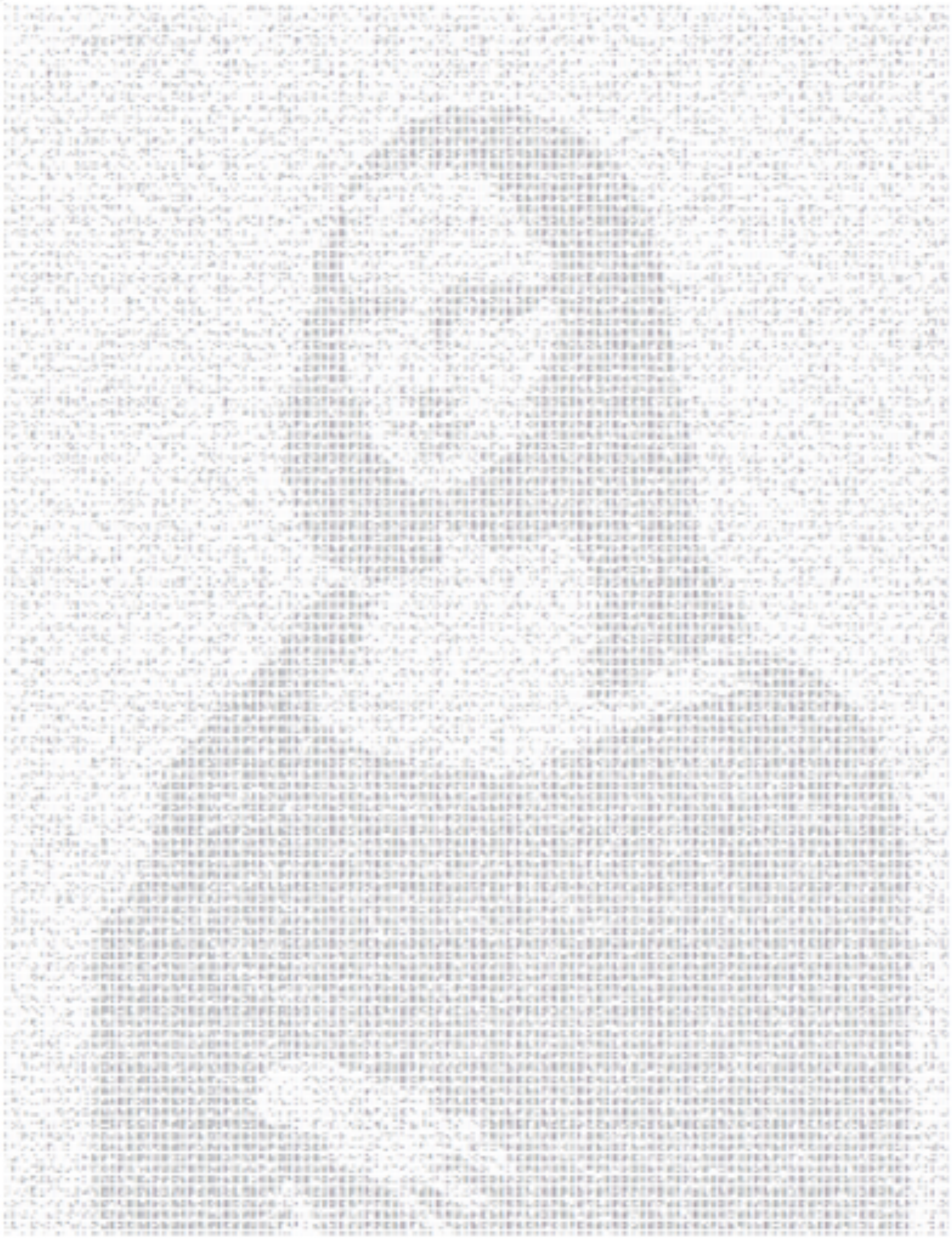
Slides adapted from Ulfar Erlingsson

An Individual's Training Data

Each bit is flipped with probability 25%

```
.......M........MM.M.......MMM.M..
......................MM...MMMM...
....M..MM.MM..MMM.M.MM.M...M..MM..
.MM......MMM....MMMMMMMMM...M...MM
..M....M........MM..MMMMMMM...M...
M.......M..MM.MMMMMMMMMMMMMMM....M
.....M.....M.M.M.MMMMMM...MMMMM...
...M.....M.MM.M.MM..M..M..MM.MMMMM
M...M.M.....M.M..M..MMM.MMMMM.MMMM
.MMM.M....M.M.M........MMMMMMMMM.M
```

Big Picture Remains!



Slides adapted from Ulfar Erlingsson

How to train a model with SGD?

```
Initialize parameters θ

For t = 1..T do
  Sample batch B of training examples
  Compute average loss L on batch B
  Compute average gradient of loss L wrt parameters θ



  Update parameters θ by a multiple of gradient average
```

How to train a model with differentially private SGD?

```
Initialize parameters θ

For t = 1..T do
  Sample batch B of training examples
  Compute per-example loss L on batch B
  Compute per-example gradients of loss L wrt parameters θ
  Ensure L2 norm of gradients < C by clipping
  Add Gaussian noise to average gradients (as a function of C)
  Update parameters θ by a multiple of noisy gradient average
```

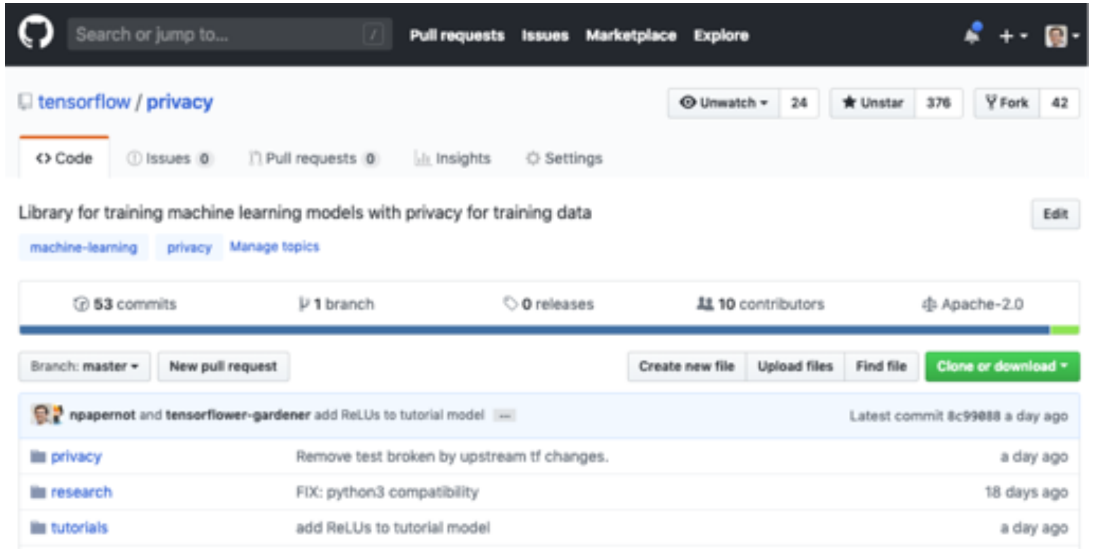
Deep Learning with Differential Privacy (CCS, 2016)
Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang

Differentially Private Stochastic Gradient Descent

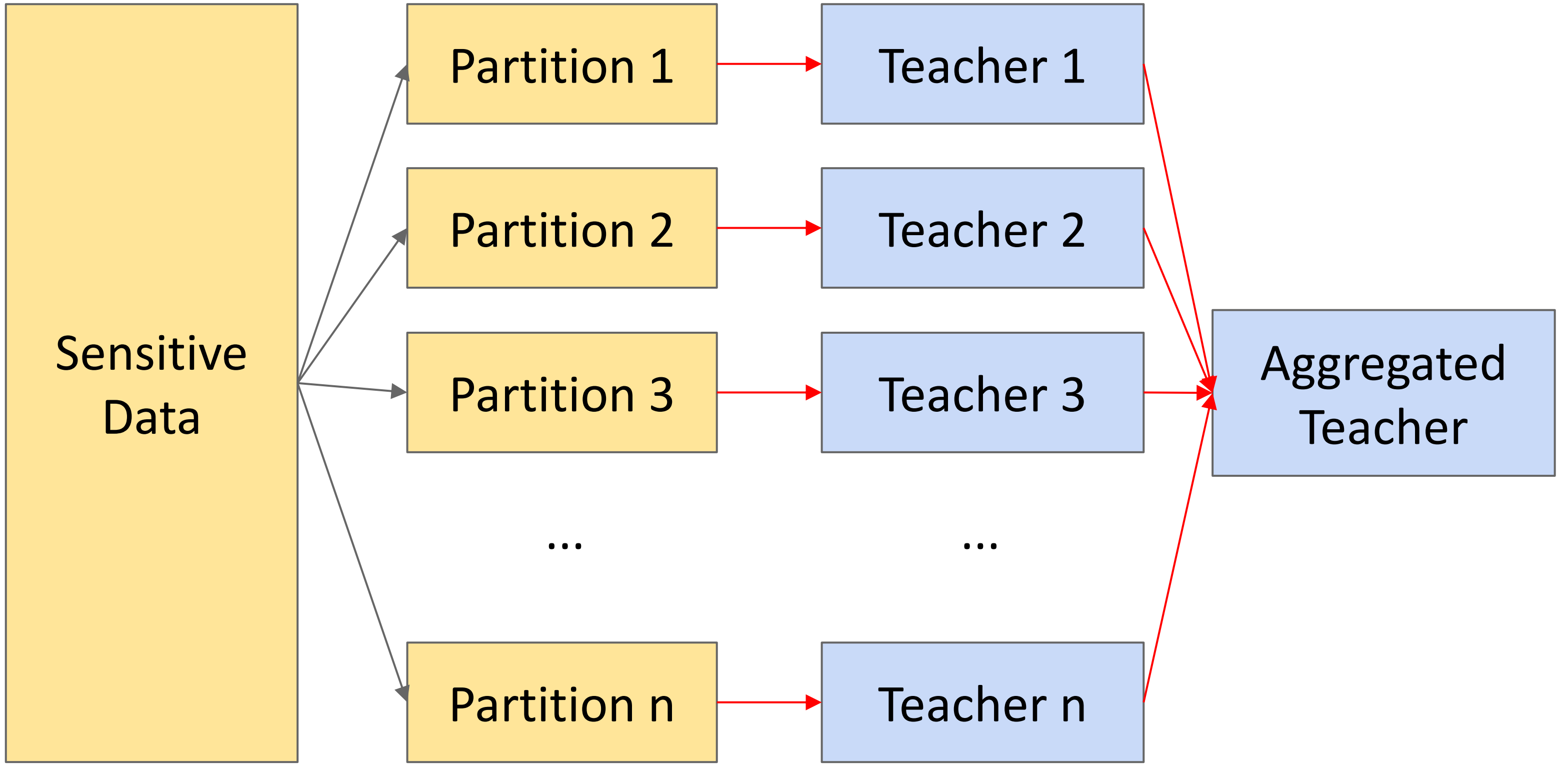
```
optimizer = tf.train.GradientDescentOptimizer(
        learning_rate=FLAGS.learning_rate)
```

↓

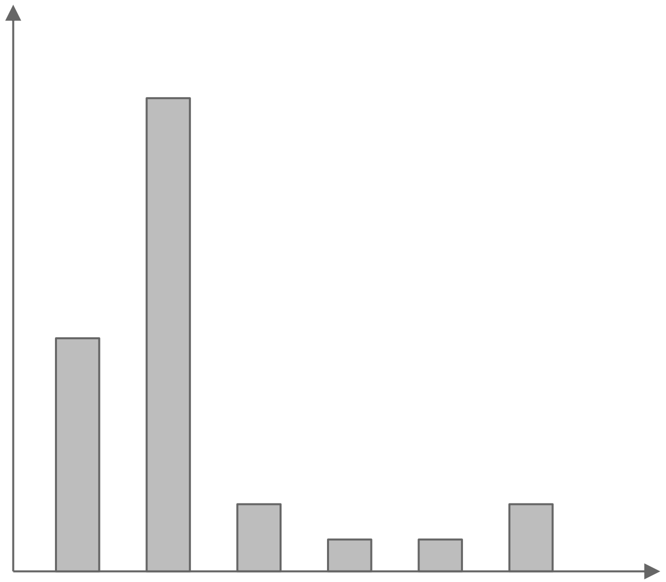
```
optimizer = VectorizedDPSGD(
        l2_norm_clip=FLAGS.l2_norm_clip,
        noise_multiplier=FLAGS.noise_multiplier,
        learning_rate=FLAGS.learning_rate)
```



PATE: Private Aggregation of Teacher Ensembles

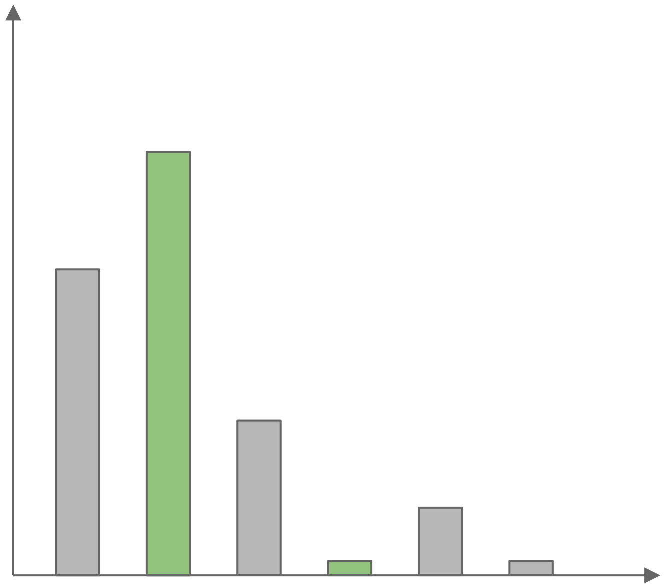


PATE: Private Aggregation of Teacher Ensembles



Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$

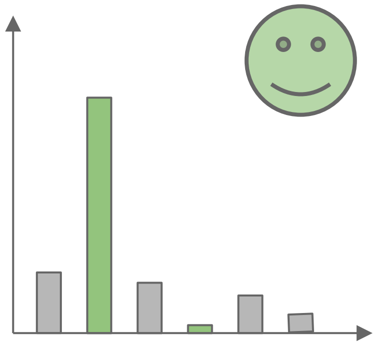


Take maximum

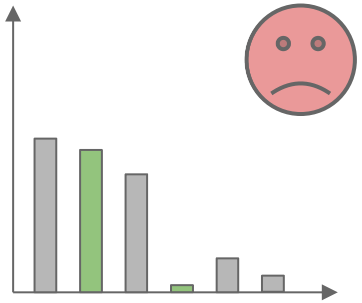
$$f(x) = \arg\max_j \left\{ n_j(\vec{x}) \right\}$$

PATE: Private Aggregation of Teacher Ensembles

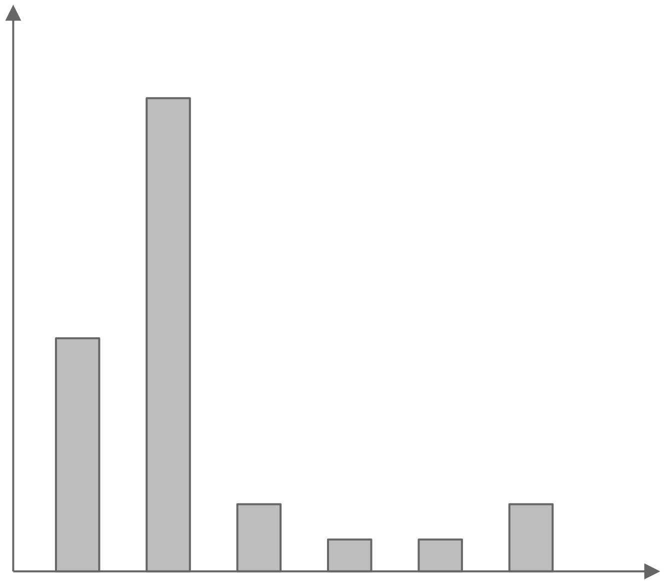
If most teachers agree on the label,
it does not depend on specific partitions,
so the privacy cost is small.



If two classes have close vote counts,
the disagreement may reveal private information.

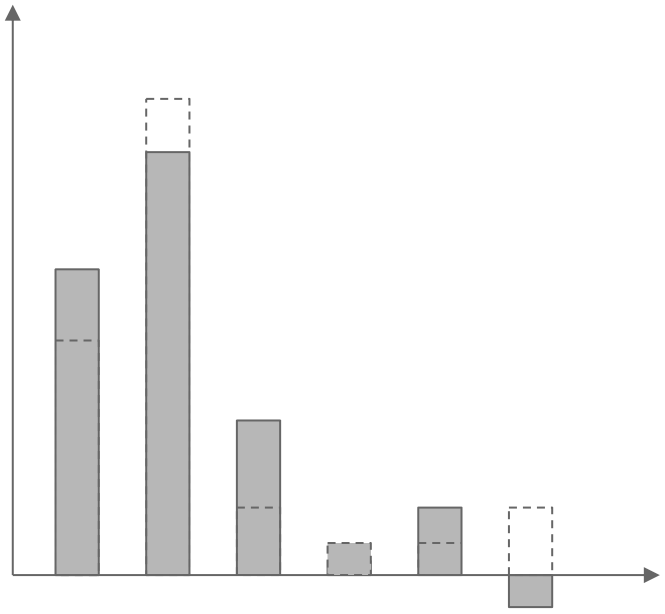


PATE: Private Aggregation of Teacher Ensembles



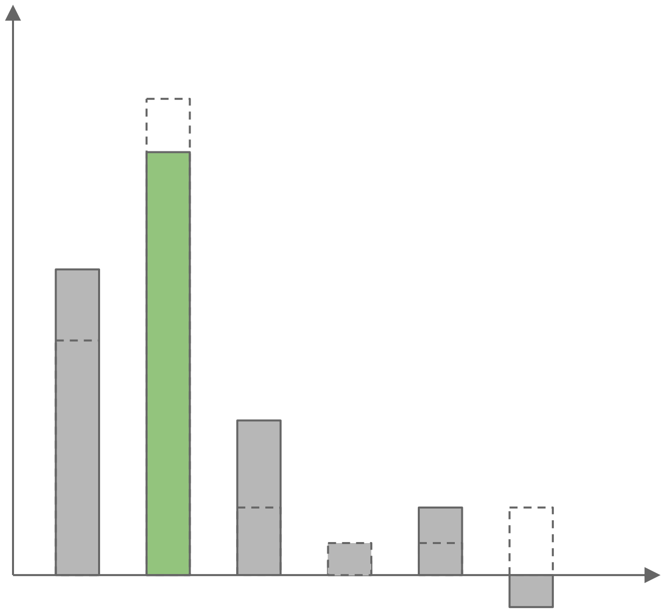
Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$



Add Laplacian

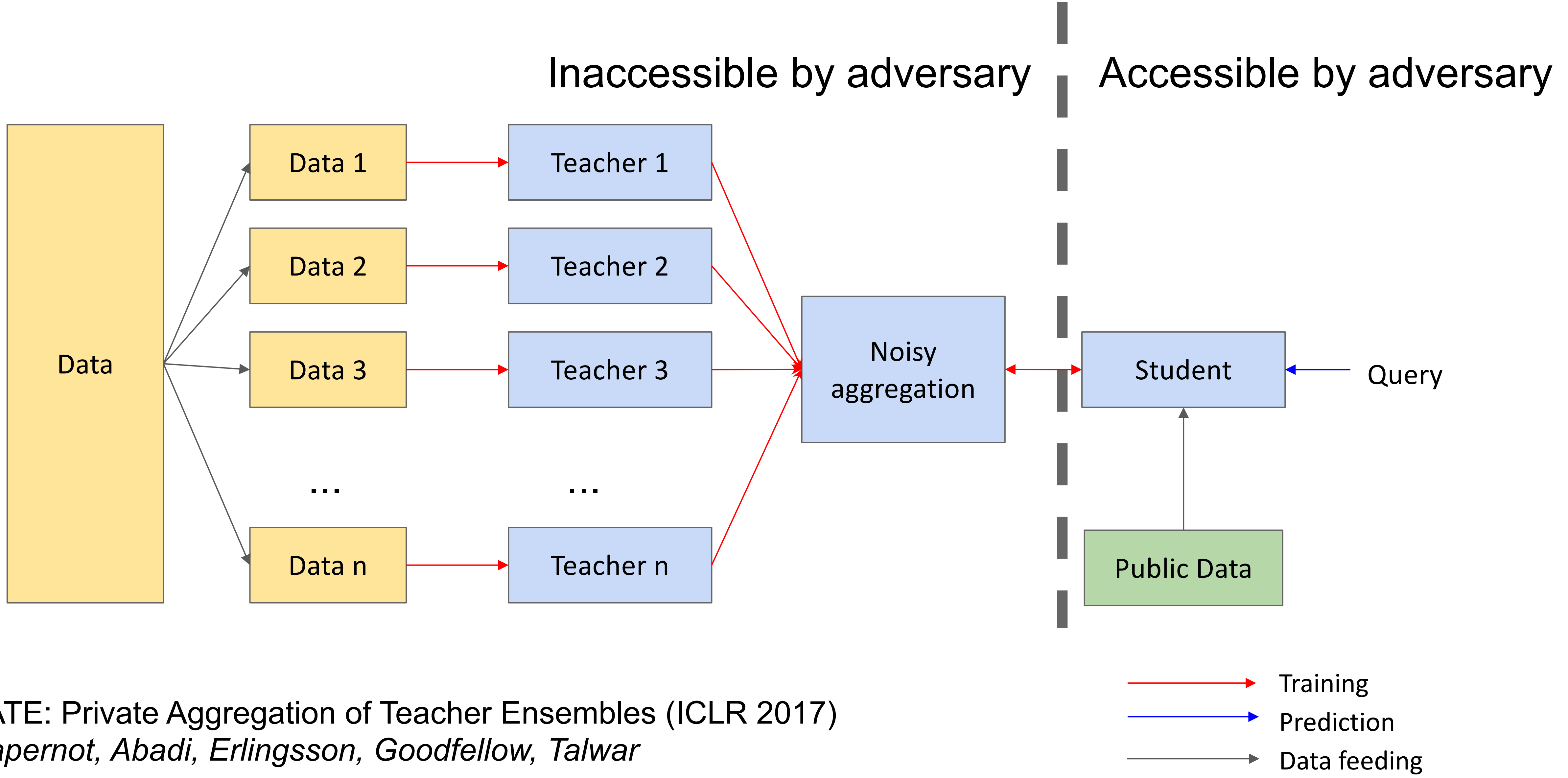
$$Lap\left(\frac{1}{\varepsilon}\right)$$



Take maximum

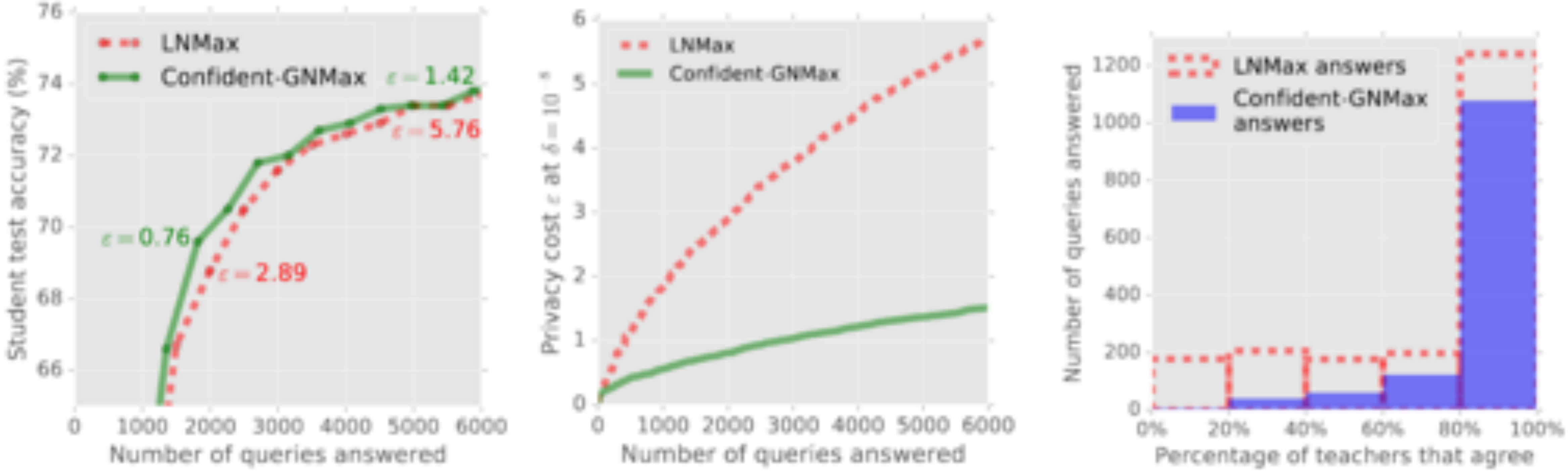
$$f(x) = \arg\max_j \left\{ n_j(\vec{x}) + Lap\left(\frac{1}{\varepsilon}\right) \right\}$$

PATE: Private Aggregation of Teacher Ensembles



PATE: Private Aggregation of Teacher Ensembles (ICLR 2017)
Papernot, Abadi, Erlingsson, Goodfellow, Talwar

Aligning privacy with generalization



Scalable Private Learning with PATE (Papernot*, Song* et al., ICLR 2018)

Model assurance and admission control

Model assurance and admission control

Machine learning objective: average-case performance
→ **Testing**

Security objective: worst-case performance
→ **Verification**

Membership inference (Shokri et al.), Data Provenance (Song & Shmatikov)
↓
Differential privacy analysis

Model assurance. (training time)
Establish with confidence that system matches security requirements.

Admission control. (test time)
Do we admit an answer for a given input into our pool of answers?
Combine input validation and sandboxing techniques.

How to specify policies for ML security & privacy?

Security

Informal security policy: learning system accurately models *exactly* the end task which the system was designed to solve.

→ Correct implementation (e.g., no numerical instabilities)
→ Solves the end task (e.g., correct predictions on all valid inputs)
→ Only solves the end task (e.g., no backdoor or other poisoned data)

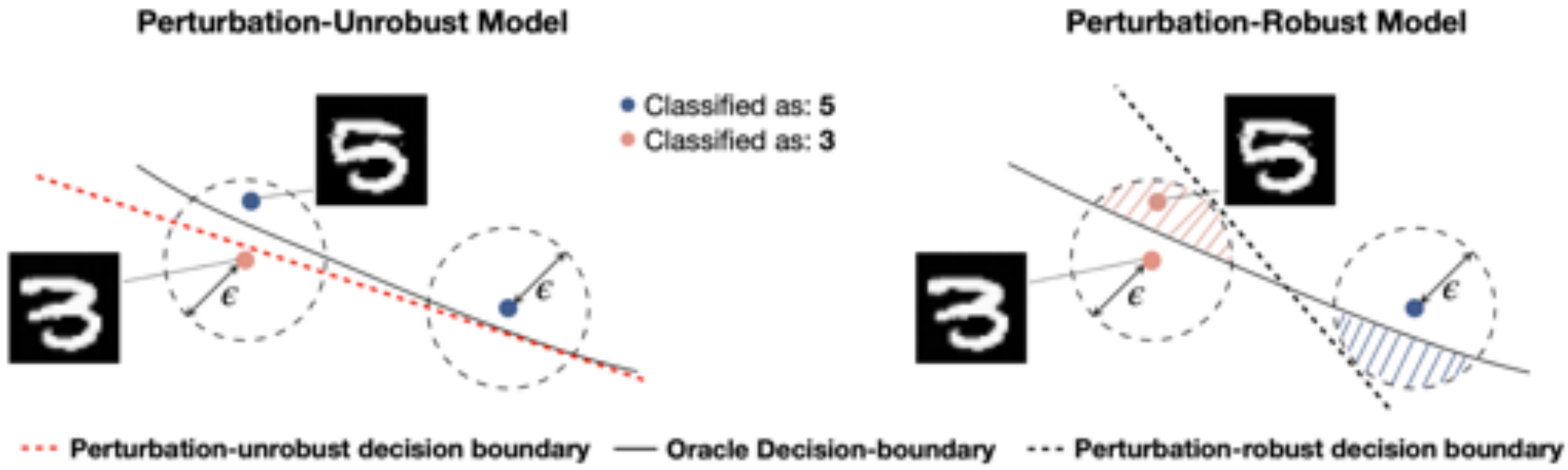
Open problem: how to formalize ML security policy with *precise semantics* while avoiding *ambiguity*?

Privacy

Privacy policy: learning behavior does not reflect any private information

Formal requirement specification: differential privacy

An example *toy* security policy: *the ℓ_p norm in vision*



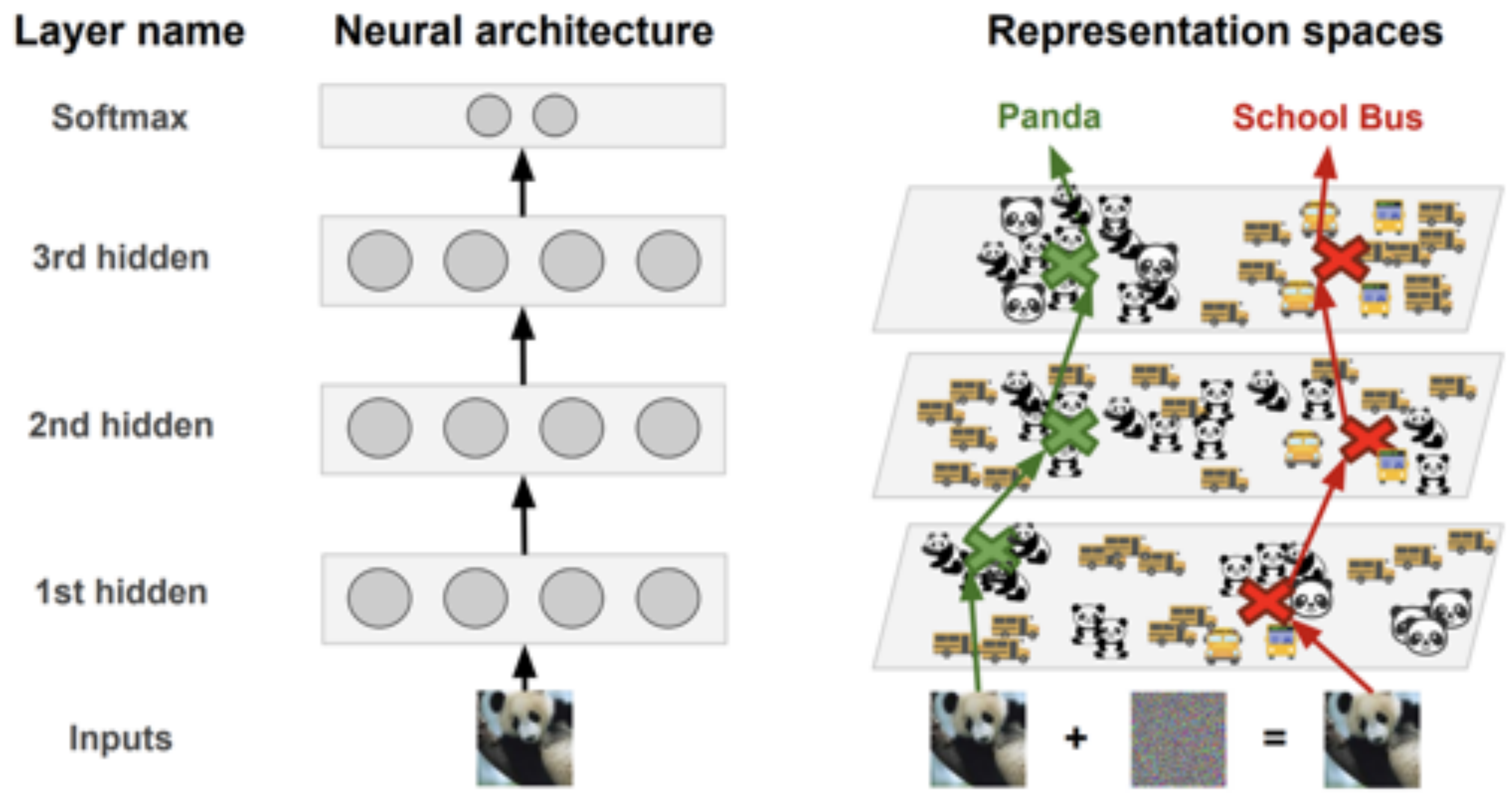
Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness (Jacobsen et al.)

Admission control at test time

Weak authentication (similar to search engines) calls for admission control:

Do we admit a sandboxed model's output into our pool of answers?

Example:
define a well-calibrated estimate of uncertainty to reject outliers (hard when distribution is unknown) through conformal prediction



Deep k-Nearest Neighbors (2018)
Papernot and McDaniel

Soft Nearest Neighbor Loss (2019)
Frosst, Papernot and Hinton

Towards auditing ML systems

The case for auditing in ML

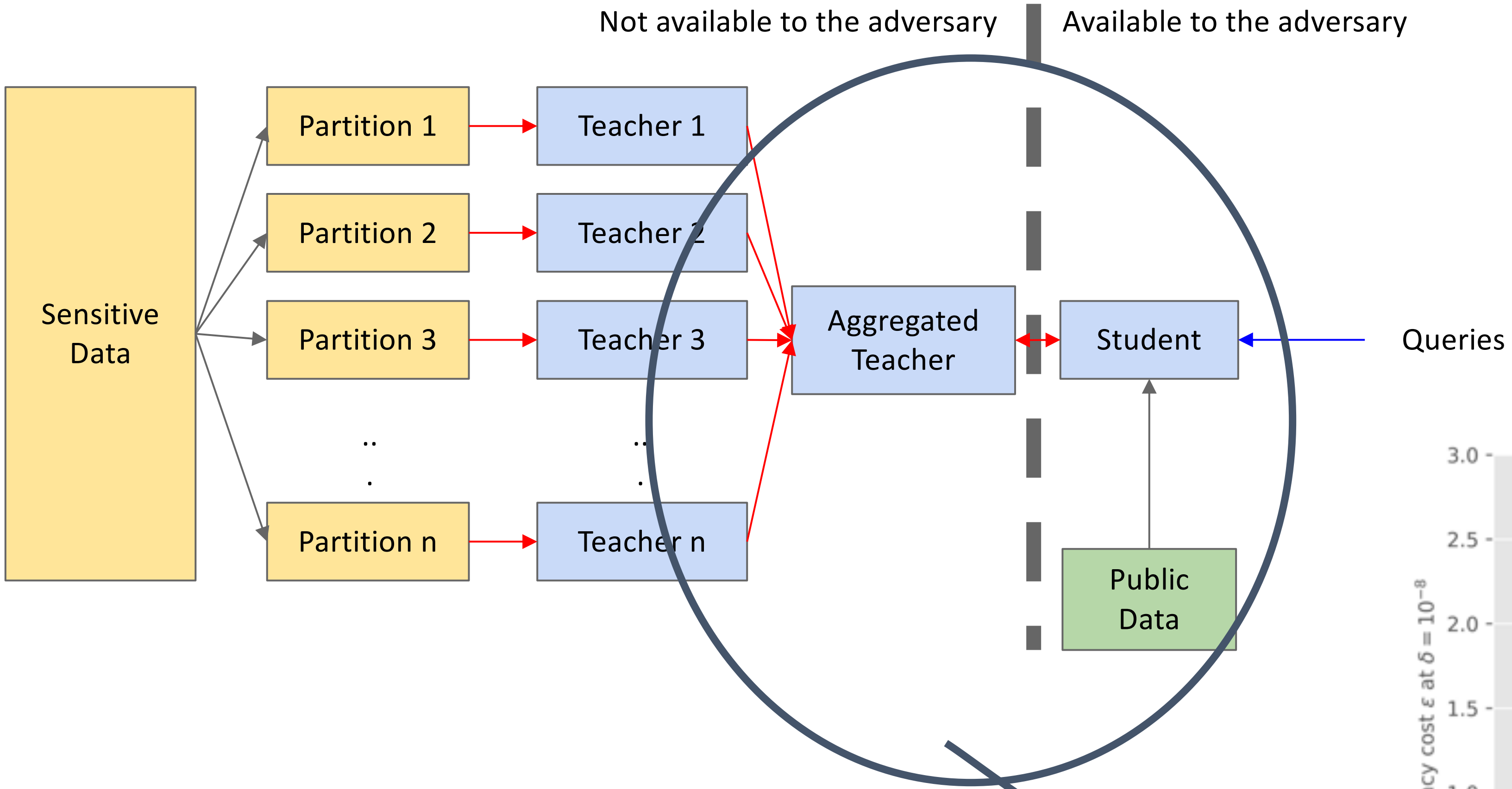
Auditing: (1) *identify* information to collect
(2) *analyze* it

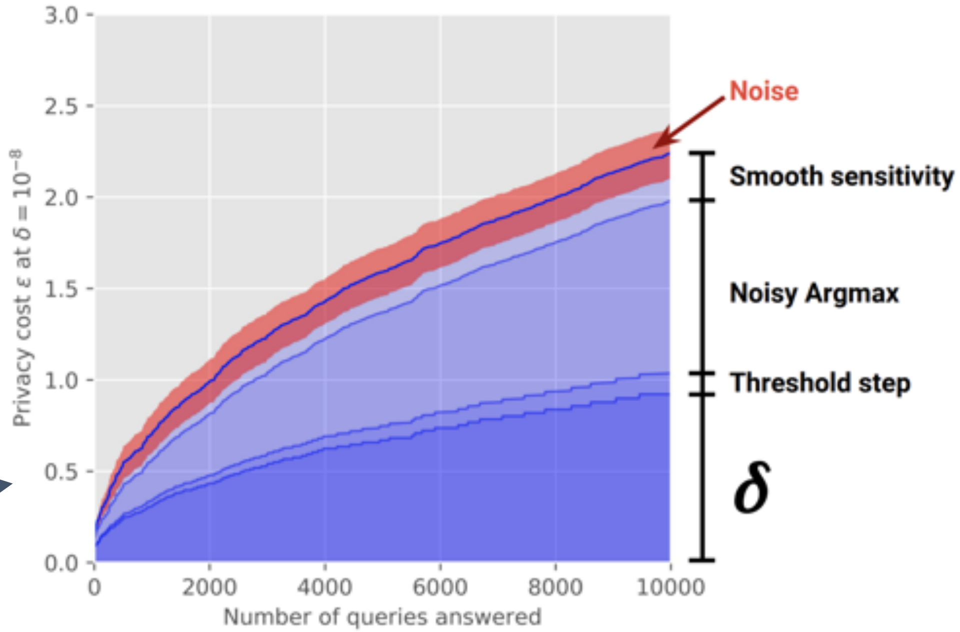
When systems have weak authentication and authorization, auditing is an important component of security. (John et al., 2010)

Auditing design is informed by specification of security policy.

Benefits: reactive and proactive identification of threats
increased work factor and psychological acceptability

Auditing the learning algorithm: an example for privacy





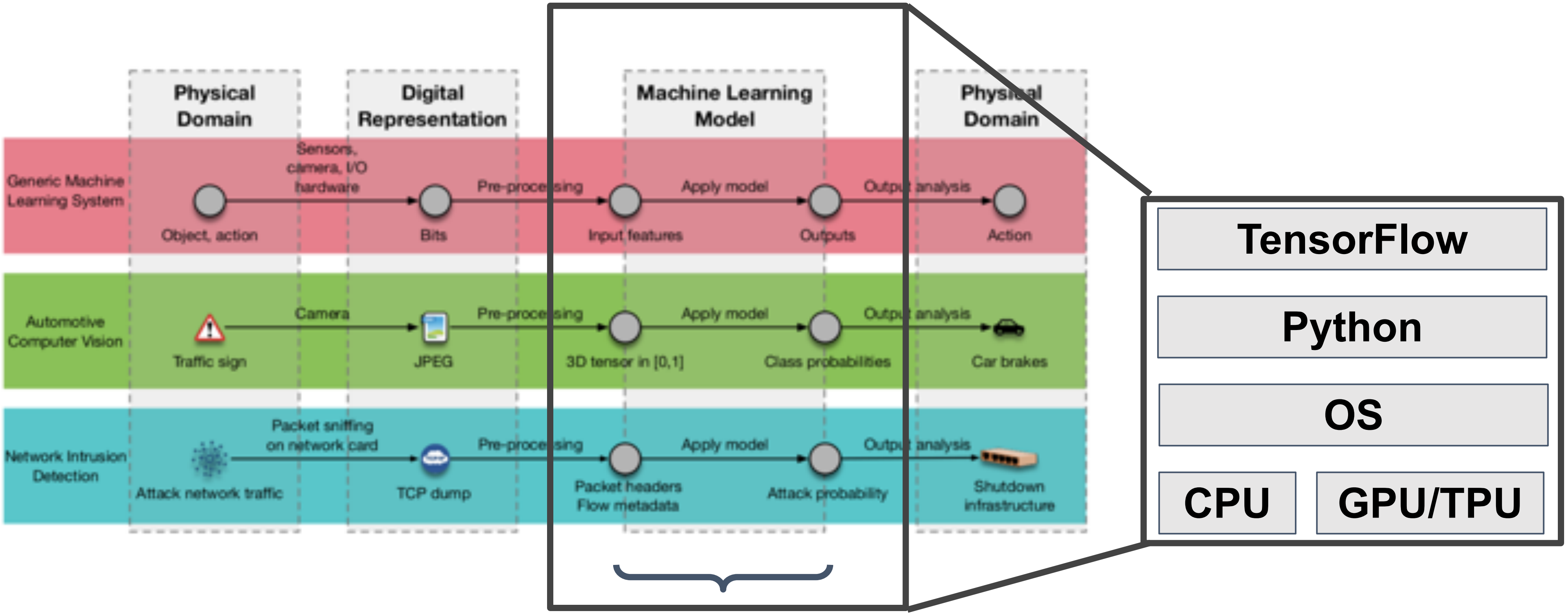
Moments accountant

arXiv:1607.00133 (Abadi et al.); arXiv:1802.08908 (Papernot*, Song* et al.); arXiv (Carlini et al.)

Conclusions

Efforts need to specify ML security and privacy policies.

What is the right abstraction and/or language to formalize security and privacy requirements with precise semantics and no ambiguity?



Towards the Science of Security and Privacy in Machine Learning (Papernot et al.)

Efforts need to specify ML security and privacy policies.

What is the right abstraction and/or language to formalize security and privacy requirements with precise semantics and no ambiguity?

Admission control and auditing may address lack of assurance.

How can sandboxing, input-output validation and compromise recording help secure ML systems when data provenance and assurance is hard?

Efforts need to specify ML security and privacy policies.

What is the right abstraction and/or language to formalize security and privacy requirements with precise semantics and no ambiguity?

Admission control and auditing may address lack of assurance.

How can sandboxing, input-output validation and compromise recording help secure ML systems when data provenance and assurance is hard?

Security and privacy should strive to align with ML goals.

How do private learning and robust learning relate to generalization? How does poisoning relate to learning from noisy data or distribution drifts?

Ressources:
cleverhans.io
github.com/tensorflow/cleverhans
github.com/tensorflow/privacy

?

Contact information:
nicolas.papernot@utoronto.ca
@NicolasPapernot