Week 11
Lecture Notes

INFS3200 Advanced Database Systems
Semester 1, 2021

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

# Data Privacy – Part 2

Lecturer: Yanjun Zhang

# + Last week review



- Privacy-preserving database publishing
  - $k$-anonymity
  - $l$-diversity
  - $t$-closeness
  - Differential privacy

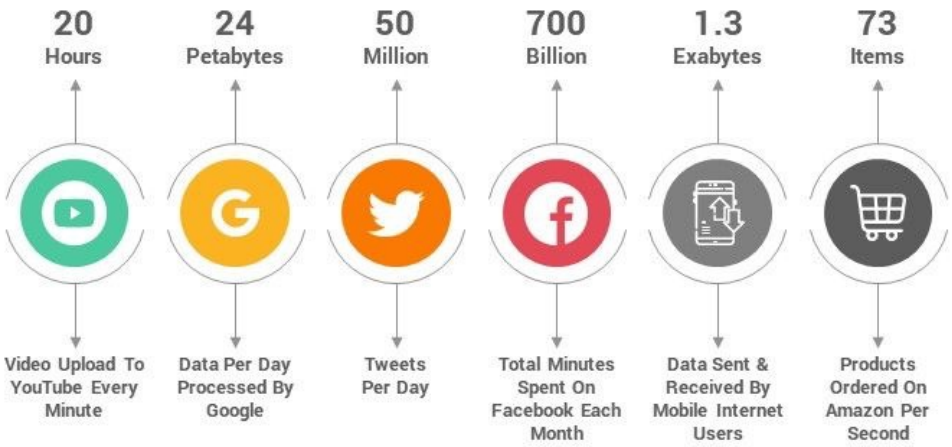
# + Outline



- Privacy and security in distributed machine learning.
  - Federated learning
  - Attacks against ML's privacy
  - Privacy-preserving sharing techniques
  - Attacks against ML's integrity and availability and corresponding defence mechanisms

# + Distributed Big Data Sharing






Healthcare

Education

Smart city

Banking and finance

# + Distributed Big Data Sharing



"The biggest obstacle to using advanced data analysis isn't skill base or technology; it's plain old access to the data."

-Edd Wilder-James, Harvard Business Review

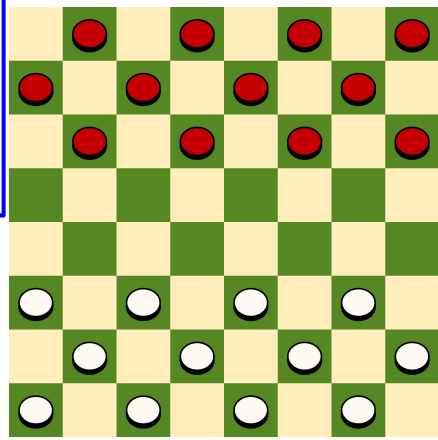
# + Machine Learning



Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.

A. L. Samuel*

**Some Studies in Machine Learning**
**Using the Game of Checkers. II—Recent Progress**



Photos from Wikipedia

# + Machine Learning - applications



- Face recognition.
- Optical character recognition: different styles, slant.
- Medical diagnosis.
- Speech recognition, machine translation, biometrics.
- Credit scoring: classify customers into high- and low-risk, based on their income and savings, using data about past loans (whether they were paid or not).

# Machine Learning with Distributed Big Data - Challenges



- ○ Data privacy
  - Data breaches that affect millions or even billions of users are far too common
  - Data protection regulations, e.g., the Australia Privacy Act and the European Union (EU) General Data Protection Regulation (GDPR)
- ○ Non-IID (independent and identical distribution)
  - The data generated by each user are quite different
- ○ Unbalanced
  - Some users produce significantly more data than others
- ○ Massively distributed
  - # mobile device owners >> avg # training samples on each device
- ○ Limited communication
  - Unstable mobile network connections

# + Federated Learning



A new paradigm – Federated Learning

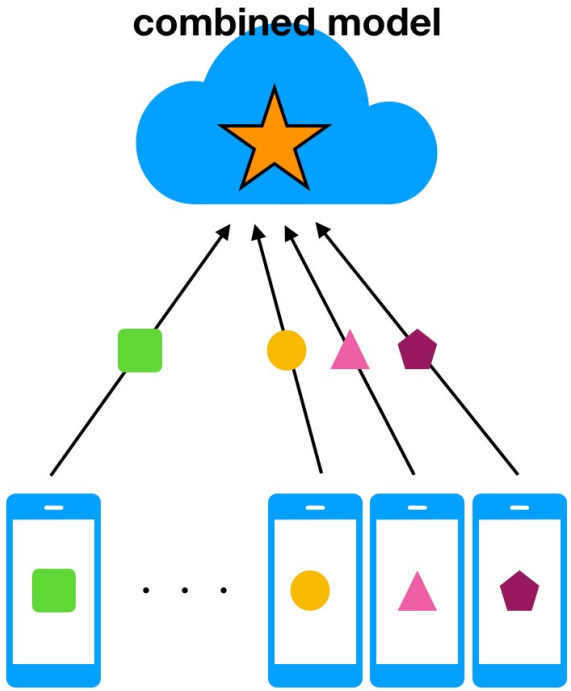
*a synchronous update scheme that proceeds in rounds of communication*

McMahan, H. Brendan, Eider Moore, Daniel Ramage, and Seth Hampson. "Communication-efficient learning of deep networks from decentralized data." *AISTATS, 2017*.

# + Federated Learning - overview



- Ensures data locality
- Reduces network communication costs
- Taps edge device computing resources




Federated learning is a great fit for smartphones, industrial and consumer IoT, healthcare and other privacy-sensitive use cases, and industrial sensor applications

# + Preliminary - Gradient Descent & Stochastic Gradient Descent (SGD)



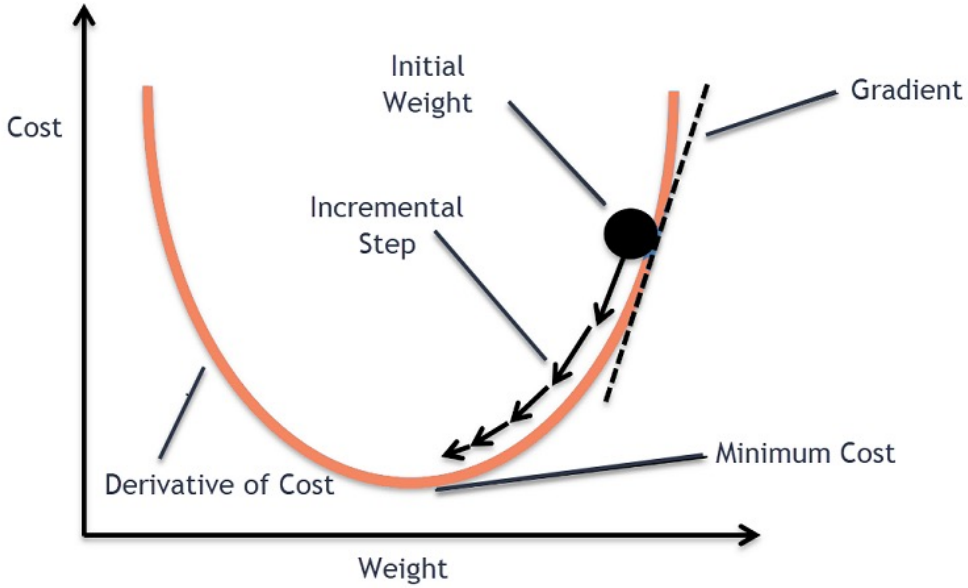
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: $\theta_0, \theta_1$

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

# + Preliminary - Gradient Descent & Stochastic Gradient Descent (SGD)




Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j = 1$ and $j = 0$)

}

***Problem**: Usually the number of training samples n is large – slow convergence*

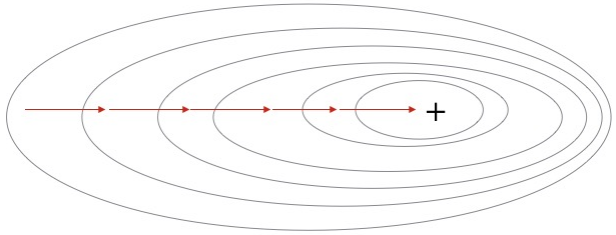
# Preliminary - Gradient Descent & Stochastic Gradient Descent (SGD)



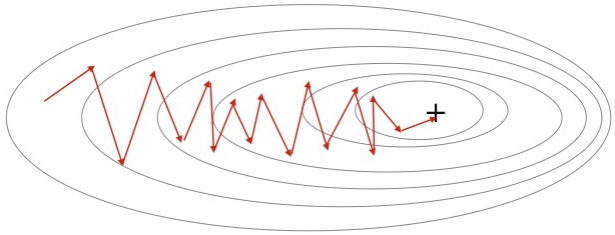
- At each step of gradient descent, instead of compute for all training samples, randomly pick a small subset (mini-batch) of training samples $(x_k, y_k)$.

- Compared to gradient descent, SGD takes more steps to converge, but each step is much faster.

Gradient Descent

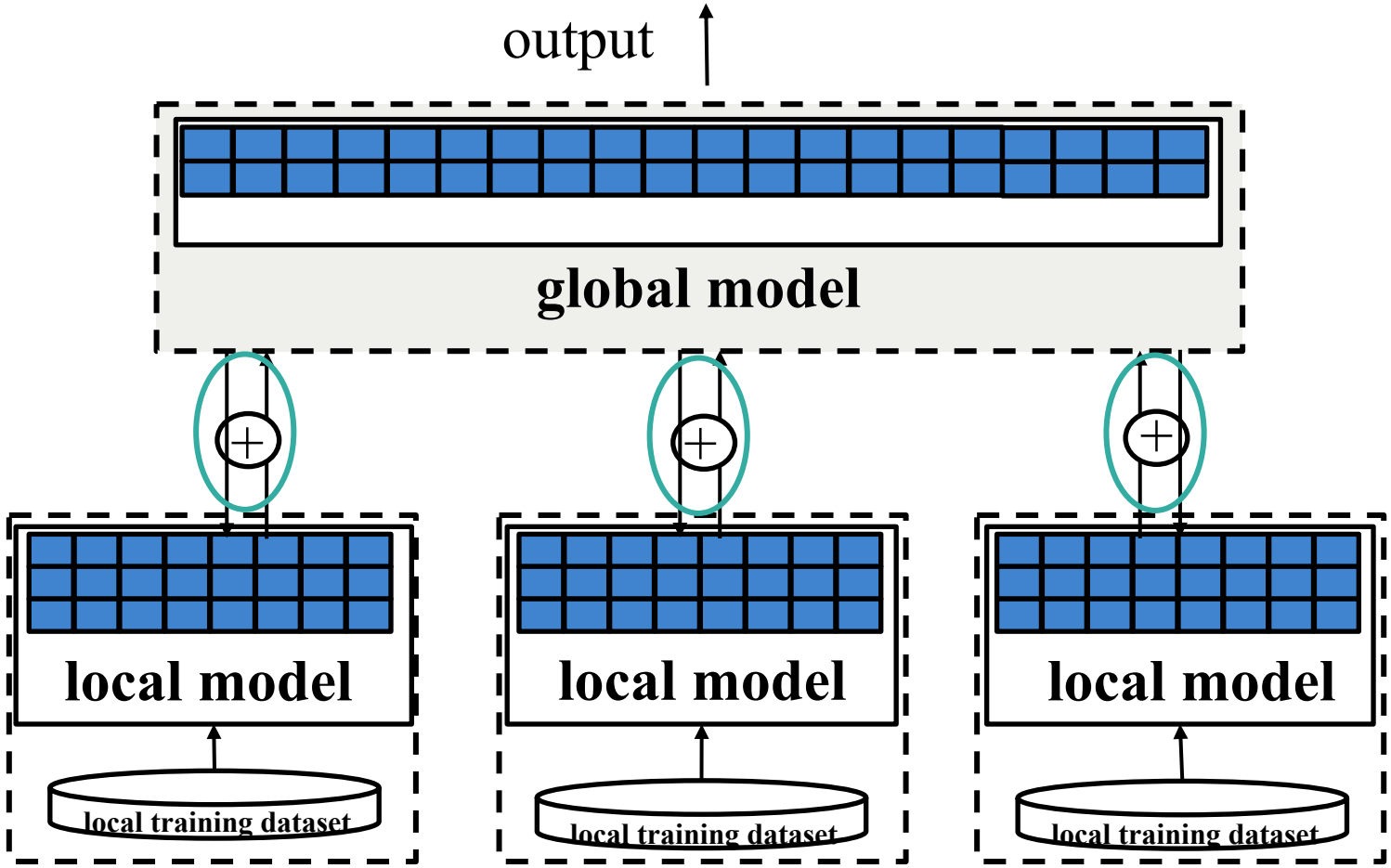


Stochastic Gradient Descent



# + Federated optimization process





# + Discussion – Privacy?



Data locality → Privacy

# + Taxonomy of Privacy Attacks Against Federated Learning

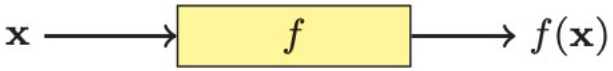


- The major criteria of the privacy attacks categorization
  - Attacker's Observation
  - Attack Mode
  - Colluding Party
  - Attack Goal

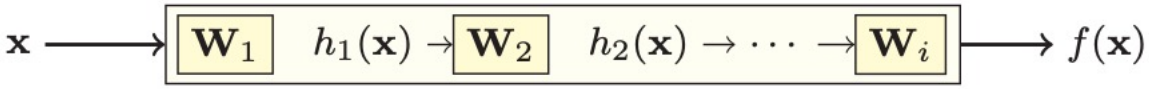
# + Attacker's Observation



- Black-box
  - The black-box adversaries are limited to interacting with the model as an oracle (e.g., by submitting inputs and observing the model's predictions).




- White-box
  - Strong white-box attackers have access to the model internals (e.g., its architecture and parameters), and any hyper-parameter that is needed to use the model for predictions. Thus, he/she can observe the intermediate computations during the training iterations.



# + Attack Mode



- Passive
  - The attacker follows the collaboration protocols and training procedures but is intended to obtain the private information of other honest nodes.
- Active
  - The attacker does not follow the protocol, who adversarially modifies his parameter uploads or sends incorrect messages to honest users.

# + Colluding Party



- Trusted aggregation node
  - The attacker can control a subset of participants who can observe the global parameter updates and can control their parameter uploads. But the aggregation node remains trusted. A trusted aggregation node can be a private server or a machine in private cloud.
- Untrusted aggregation node
  - In addition to the participants, the attacker can also control the aggregation node, who observes individual updates over time and can control the view of the participants on the global parameters.

# + Attack Goal

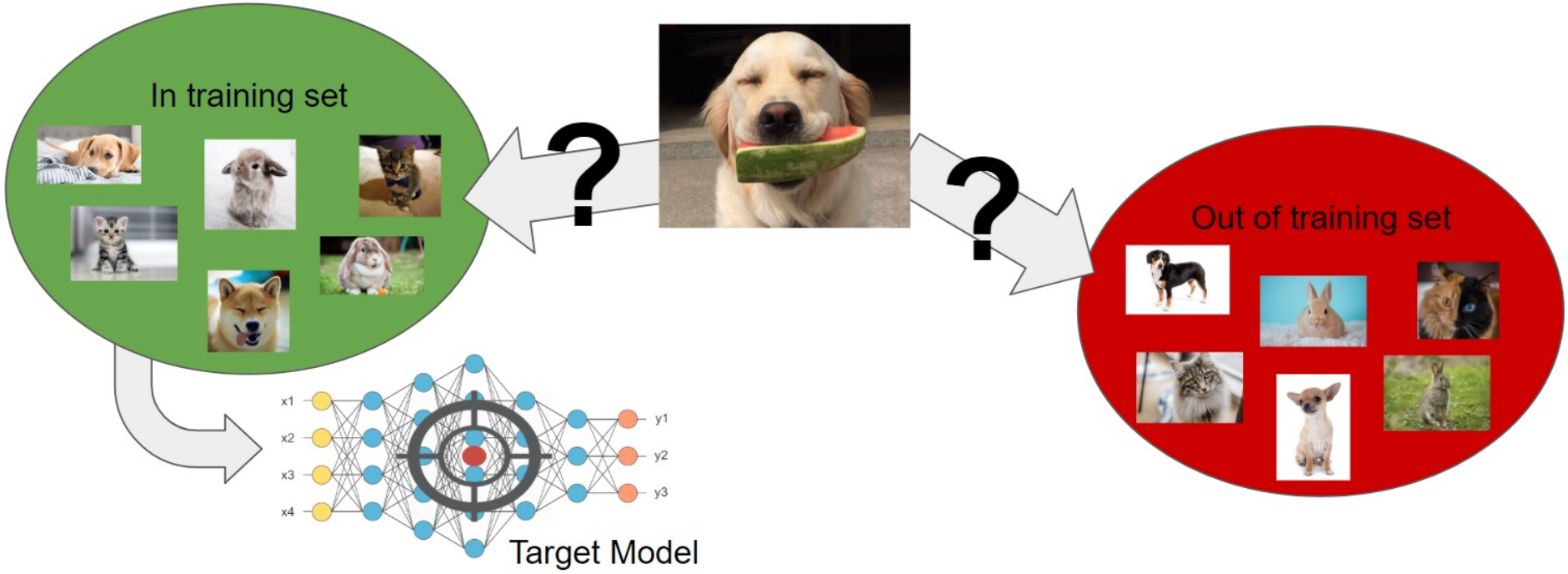


- Membership inference
  - The attacker's objective is to infer if a particular individual data record was included in the training dataset. This is a decisional problem, and its accuracy directly demonstrates the leakage of the model about its training data.
- Training data extraction
  - The attacker's objective is to infer attributes of the records in the training set.

# + Membership inference



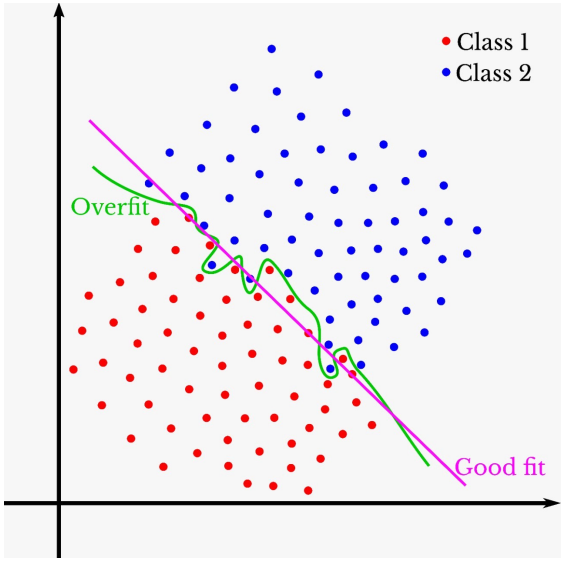
- Given a machine learning model and a record, determine whether this record was used as part of the model's training dataset or not.



# + Why it works?



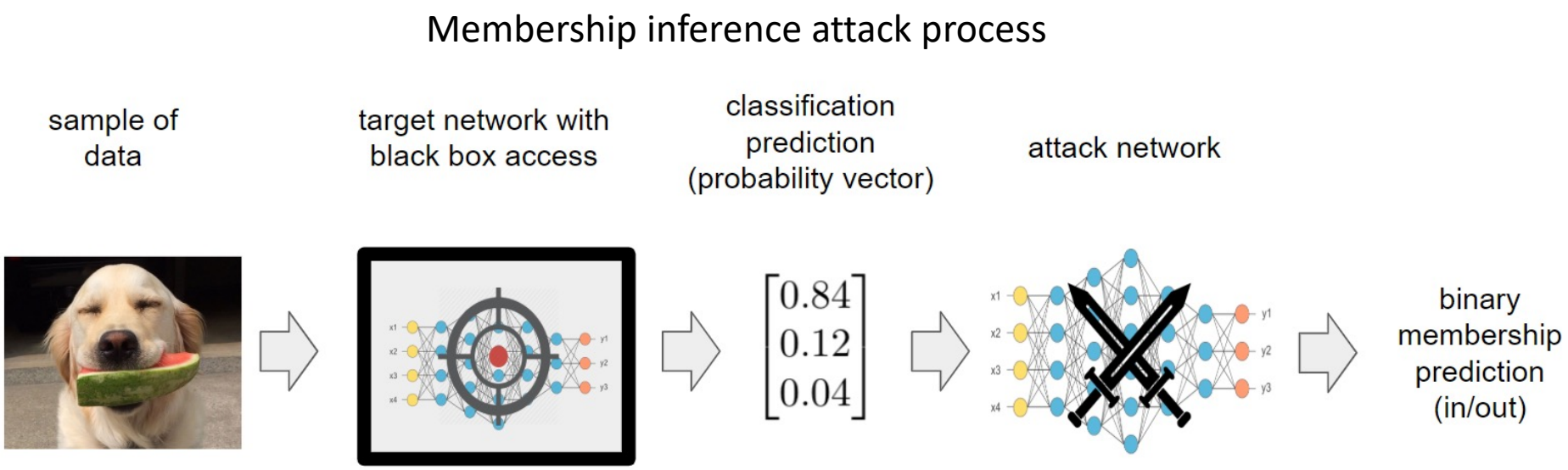
- Machine learning models respond differently to inputs which were members of the training dataset.
- These predictions from training set and non-training set data records produce two distributions that can be learned by an attack model.
- This behaviour is worse when models overfit to the training data.



# + Membership inference [Shokri,2017]



- Black-box, Passive adversary.

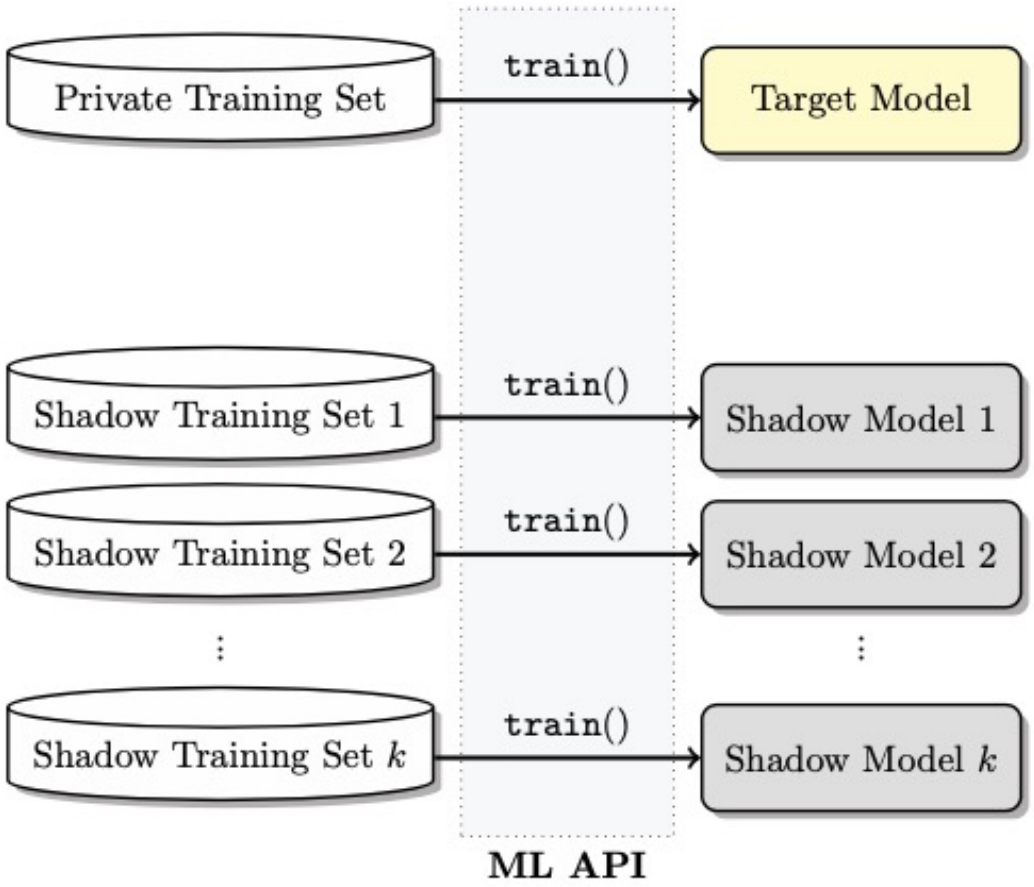



- How to construct the attack network without knowledge regarding the original training dataset of the target model?

# + Membership inference [Shokri,2017]

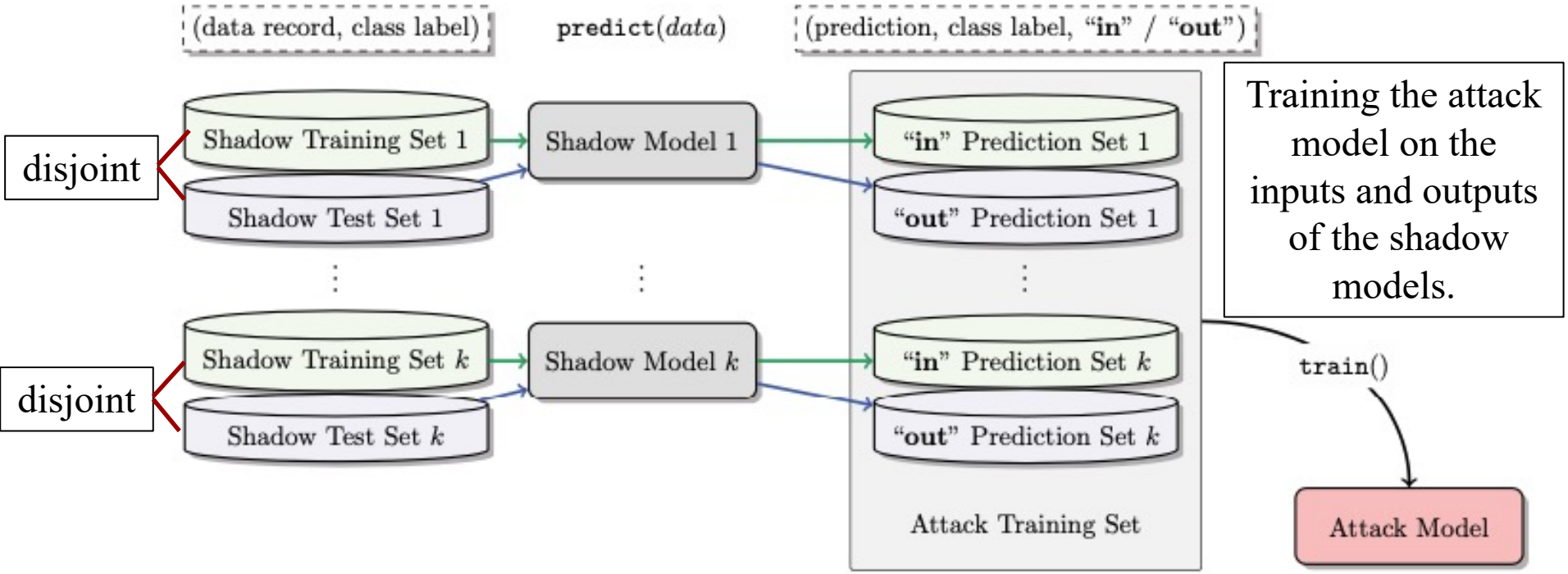


Training shadow models



# + Membership inference [Shokri,2017]





# + Membership inference [Nasr,2019]

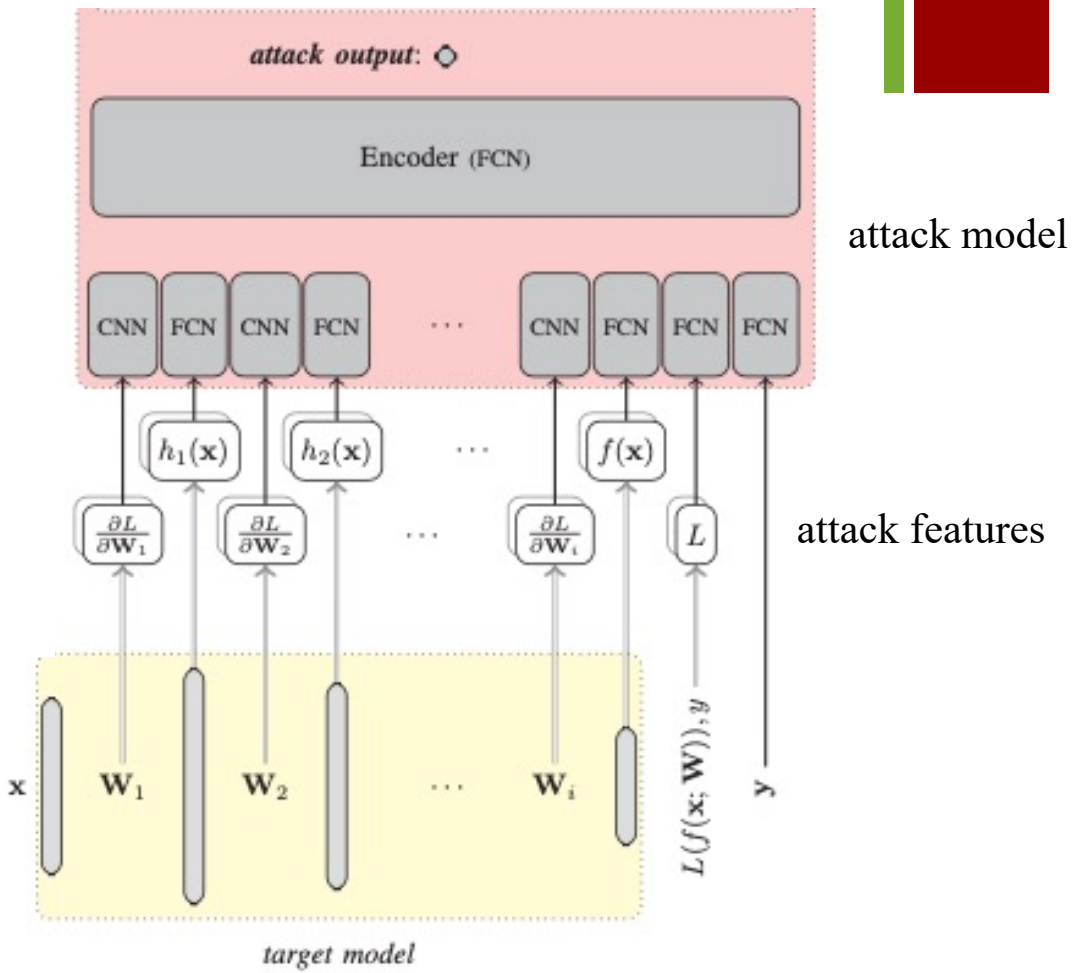


- White-box adversary
- Can be either passive or active
- Against Federated learning
  - The adversary can be the centralized parameter server;
  - or one of the participants.

# + Membership inference [Nasr,2019]



- The architecture of white-box inference attack.
  - Input features to the attack.
  - Inference attack model.



# + Membership inference [Nasr,2019]



- Active Inference Attack
  - The attacker runs a gradient *ascent* on the target x , and updates its local model parameters in the direction of increasing the loss on x.
  - If the target record x is in the training set of a participant, its local SGD algorithm abruptly reduces the gradient of the loss on x. This can be detected by the attack model and be used to distinguish members from non-members.

# + Membership inference [Nasr,2019]



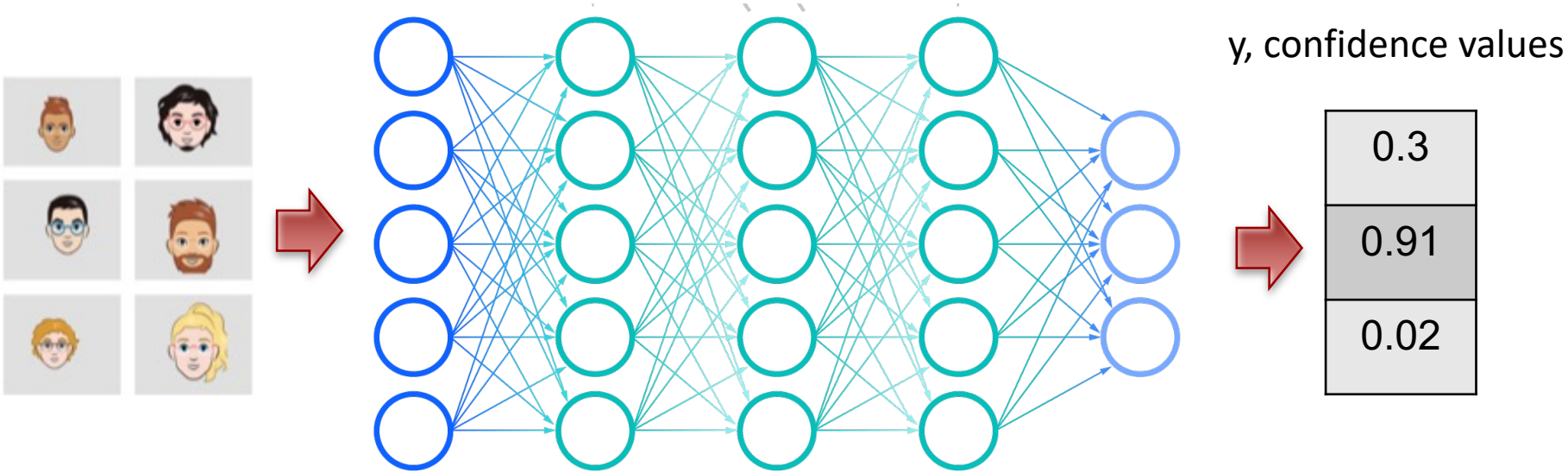
- Attack accuracy in the federated learning setting

| Target Model | | Global Attacker (the parameter aggregator) | | | | Local Attacker (a participant) | |
|---|---|---|---|---|---|---|---|
| | | Passive | Active | | | Passive | Active |
| Dataset | Architecture | | Gradient Ascent | Isolating | Isolating Gradient Ascent | | Gradient Ascent |
| CIFAR100 | Alexnet | 85.1% | 88.2% | 89.0% | 92.1% | 73.1% | 76.3% |
| CIFAR100 | DenseNet | 79.2% | 82.1% | 84.3% | 87.3% | 72.2% | 76.7% |
| Texas100 | Fully Connected | 66.4% | 69.5% | 69.3% | 71.7% | 62.4% | 66.4% |
| Purchase100 | Fully Connected | 72.4% | 75.4% | 75.3% | 82.5% | 65.8% | 69.8% |

# + Model Inversion - Training data extraction [Fredrikson, 2015]

- Can be either black-box/white-box,
- Passive adversary.

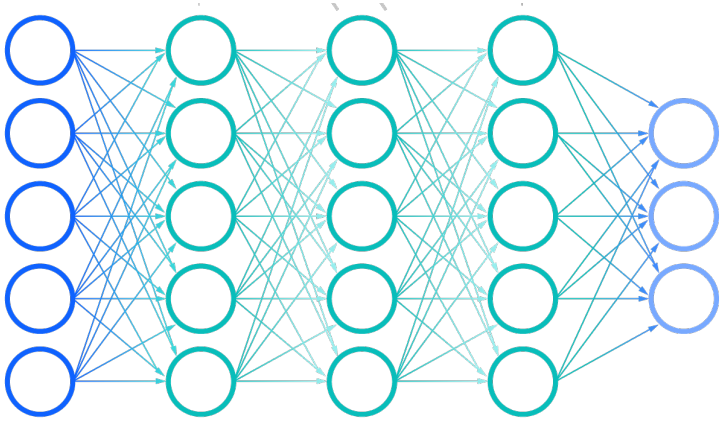
Face recognition



# + Model Inversion - Training data extraction [Fredrikson, 2015]



The adversary has access only to a trained model, but not to any of the original training data



Queries: $x$

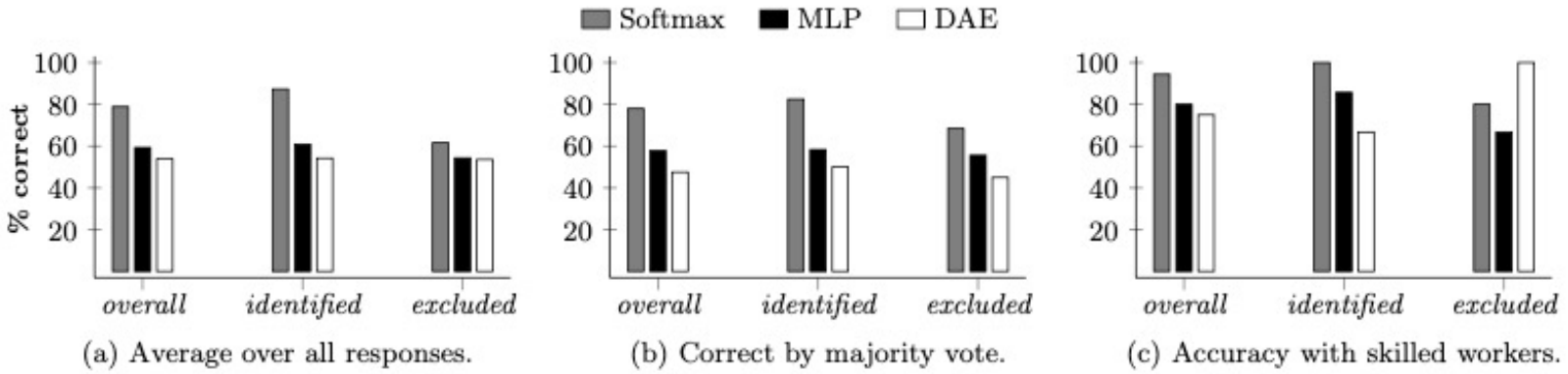
$\tilde{f}(x)$, confidence value

Minimize loss(y, $\tilde{f}(x)$)

# + Model Inversion - Training data extraction [Fredrikson, 2015]

Extracted face

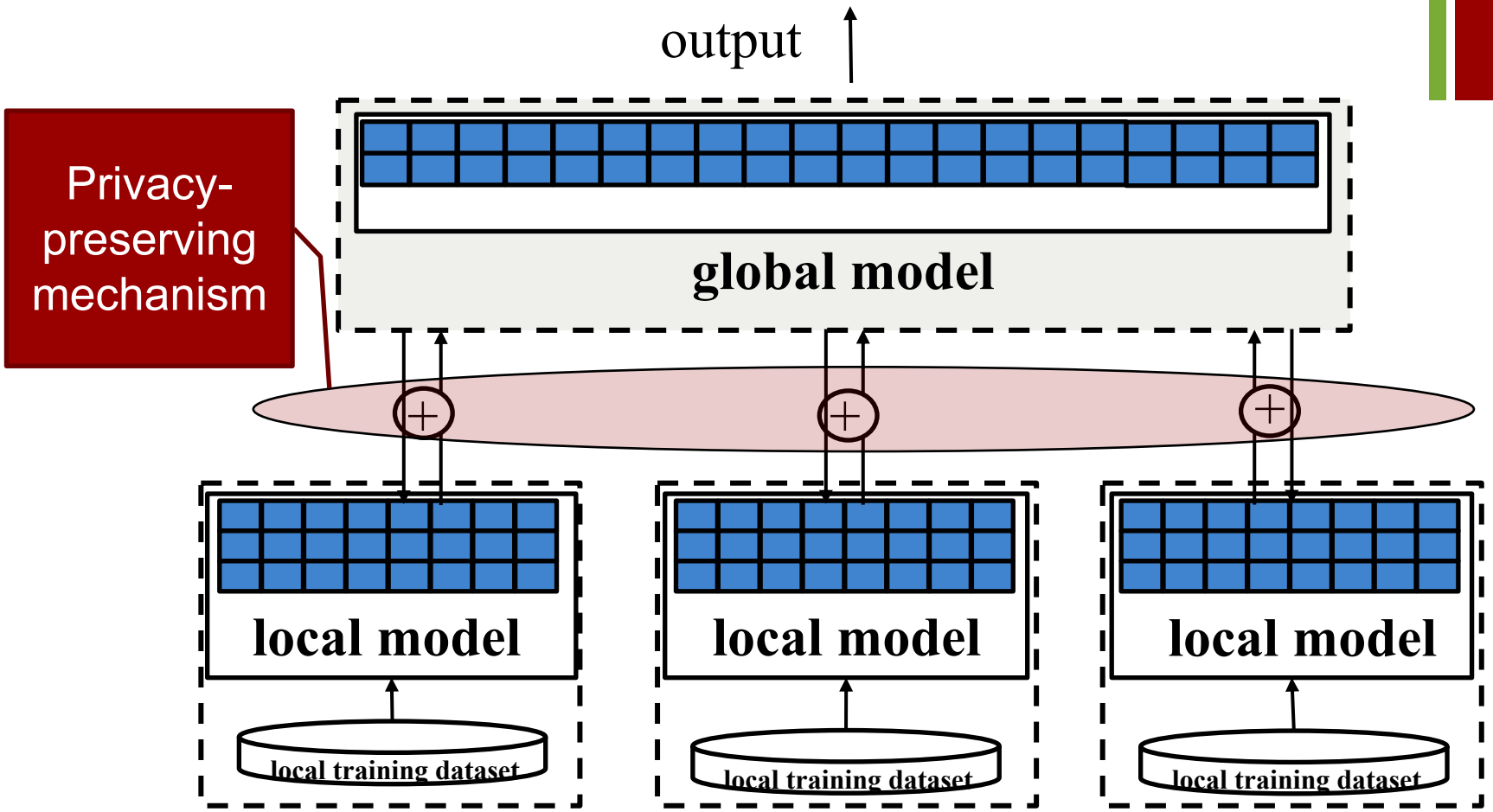
Training data



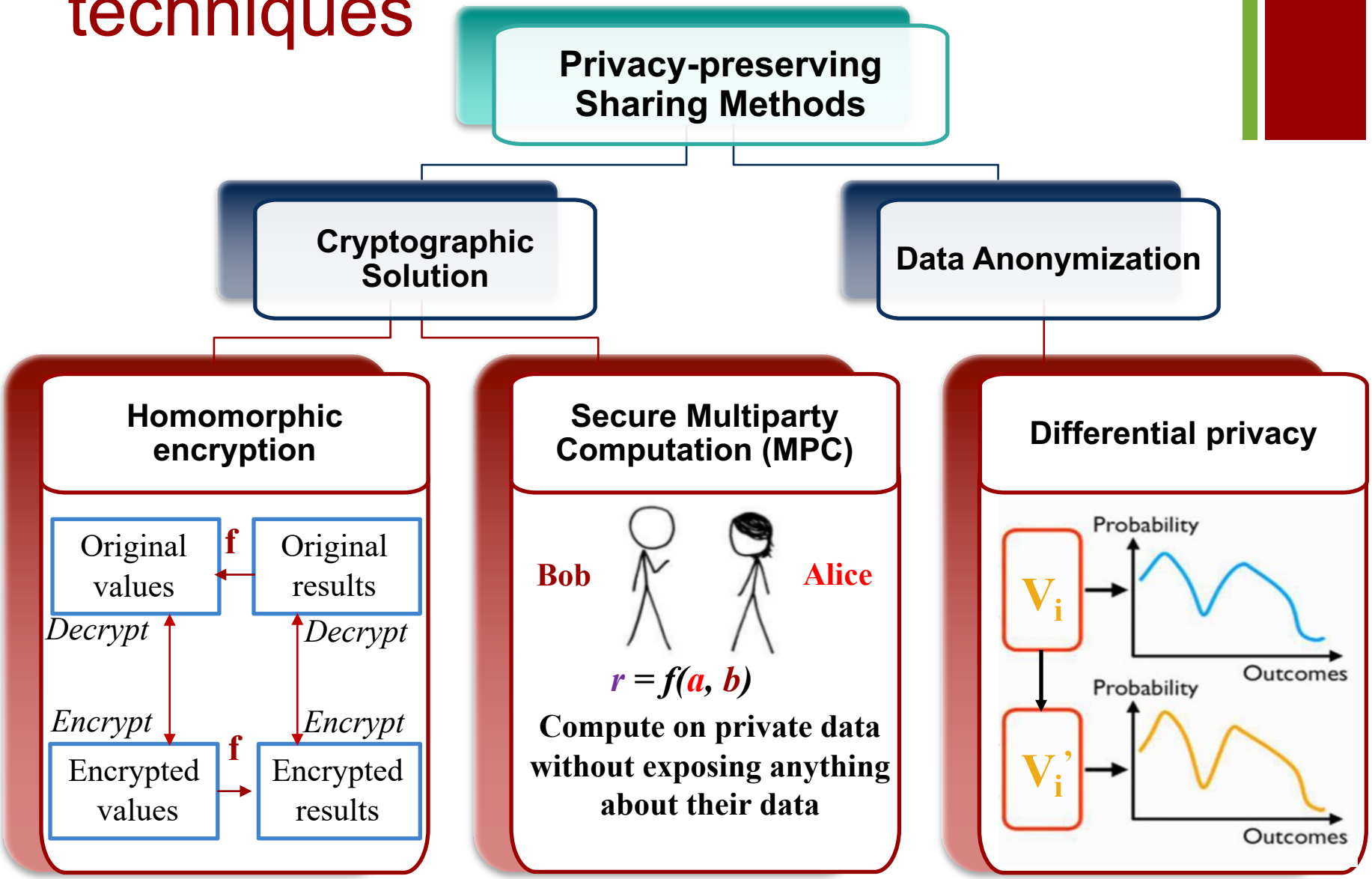

Attack results from Mechanical Turk surveys

# + Privacy-preserving Synthesis

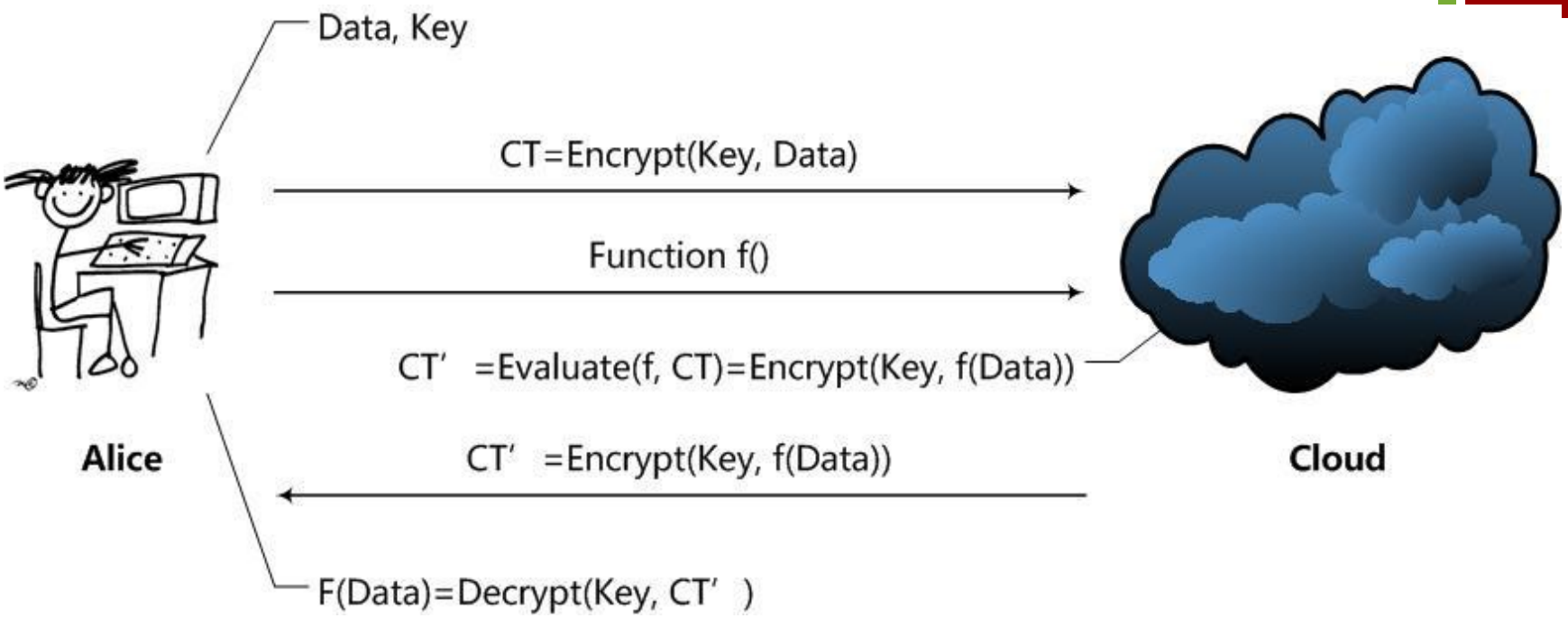




# + Privacy-preserving sharing techniques



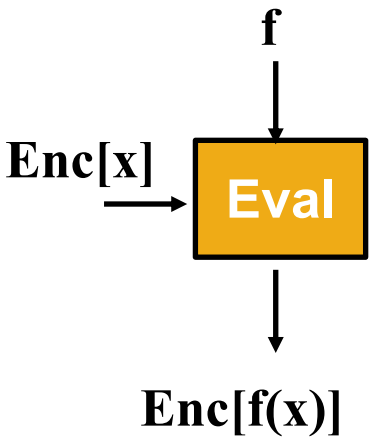
# + Homomorphic Encryption[Gentry C, 2009]



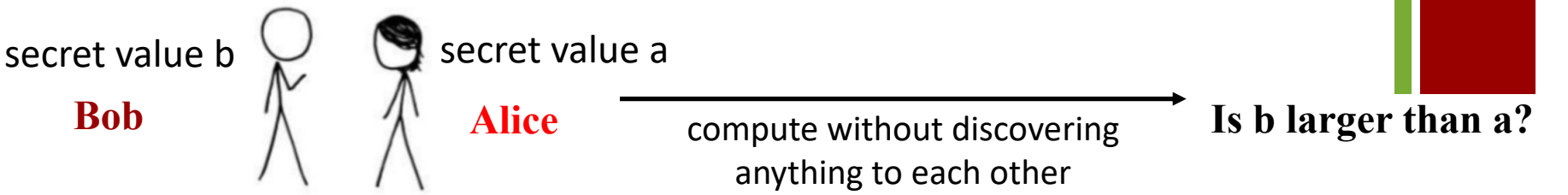
# + Homomorphic Encryption



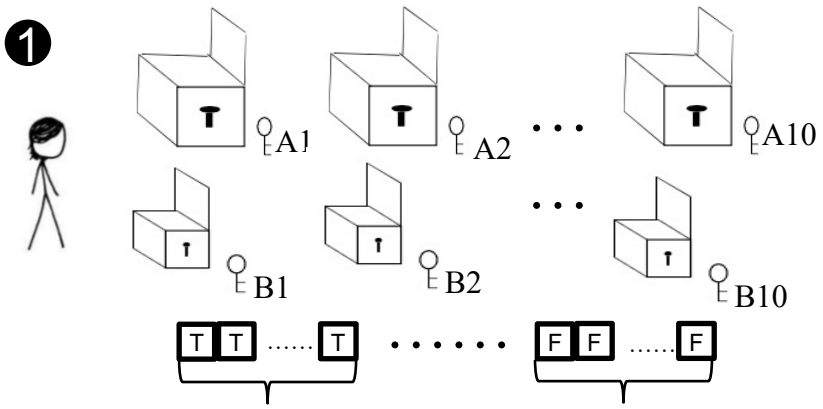
- **Fully Homomorphic Encryption**
  - Supports ***arbitrary computation*** on ciphertexts
  - Not practical (yet) – 1 kb takes 15 mins
- **Partially Homomorphic Encryption**
  - Support a limited number of accumulated operations: addition / multiplication



# + Secure Multiparty Computation [Yao, 1982]

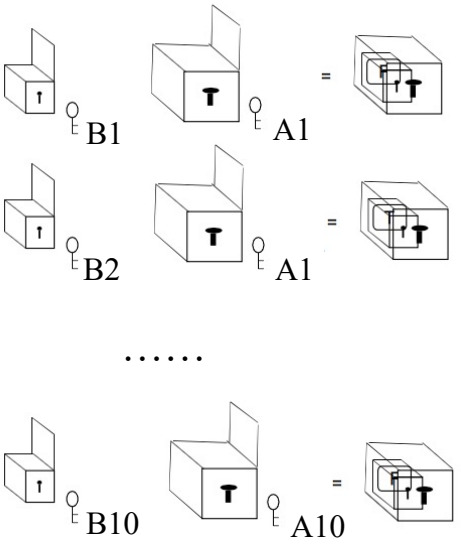



❶

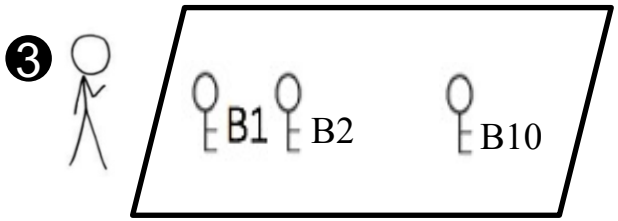



❷

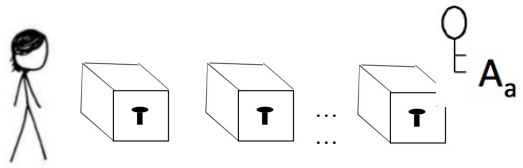
| Alice a | Bob b | f(a,b), is b larger? |
|---|---|---|
| 1 | 1 | F |
| 2 | 1 ..... | F |
| | | |
| 1 | 2 ..... | T |
| | | |
| 10 | 10 | F |



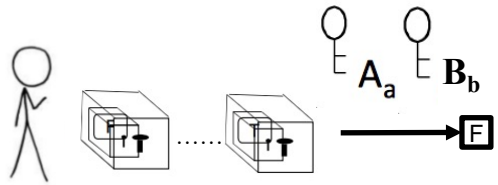

❸




❹

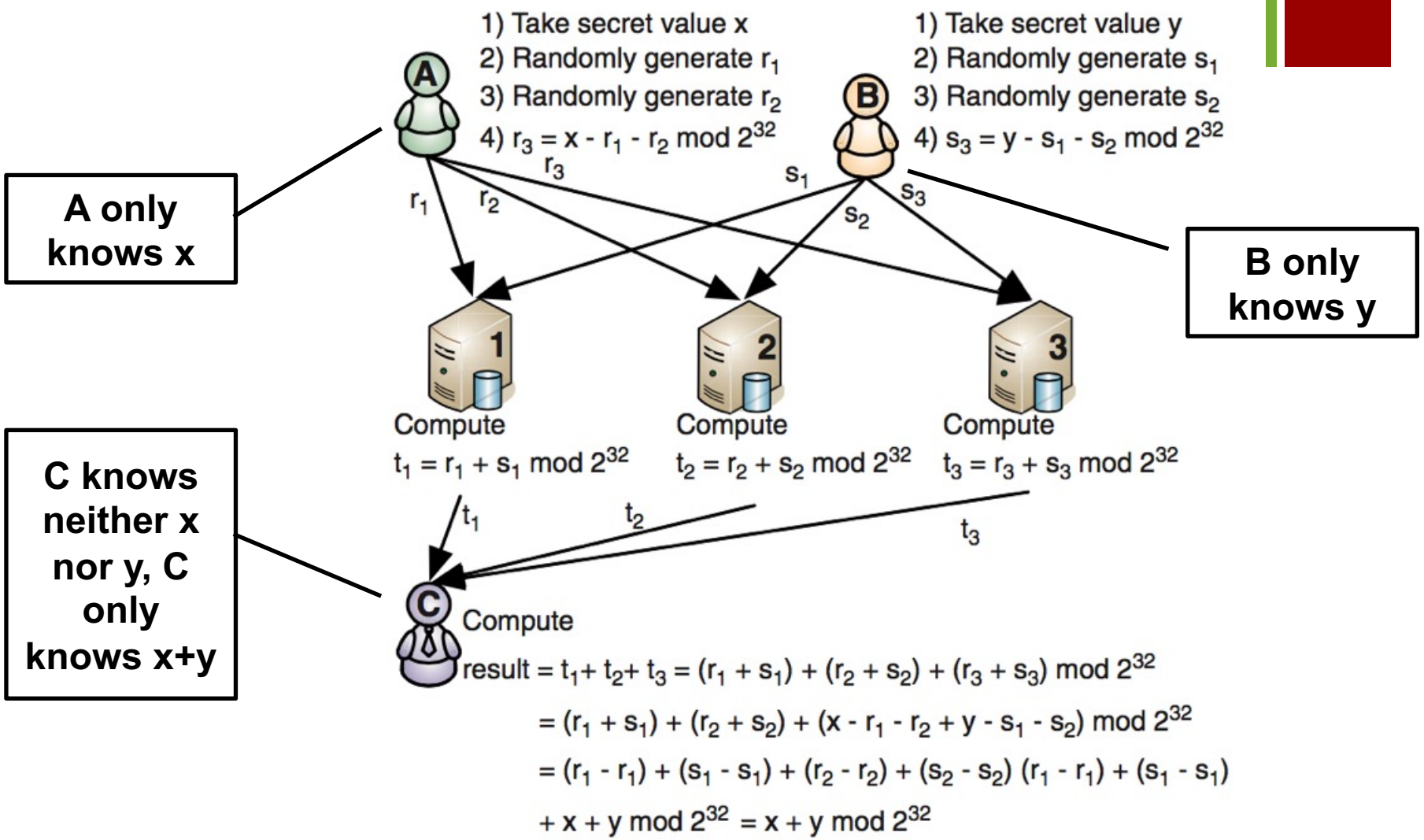



❺

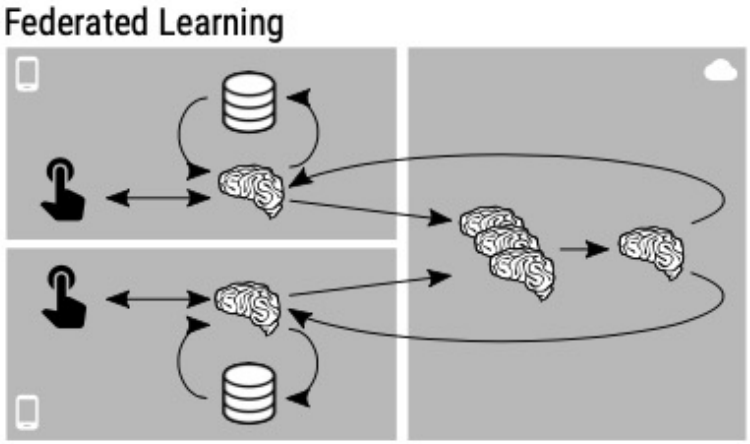


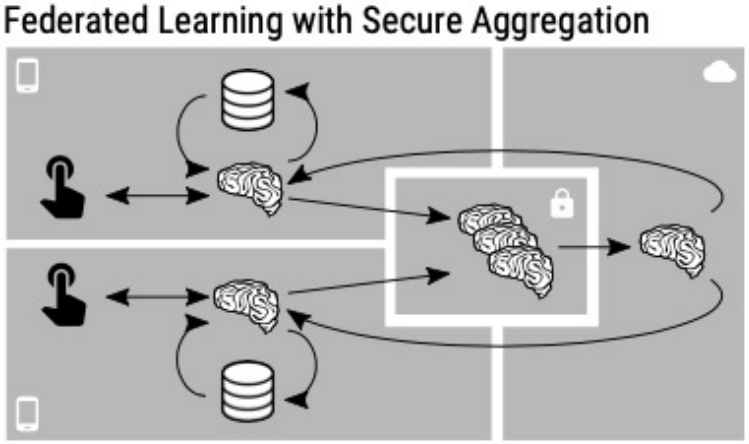
# + MPC for addition [Kamm, 2013]





# + Securely Aggregated Federated Learning [Bonawitz, 2017]

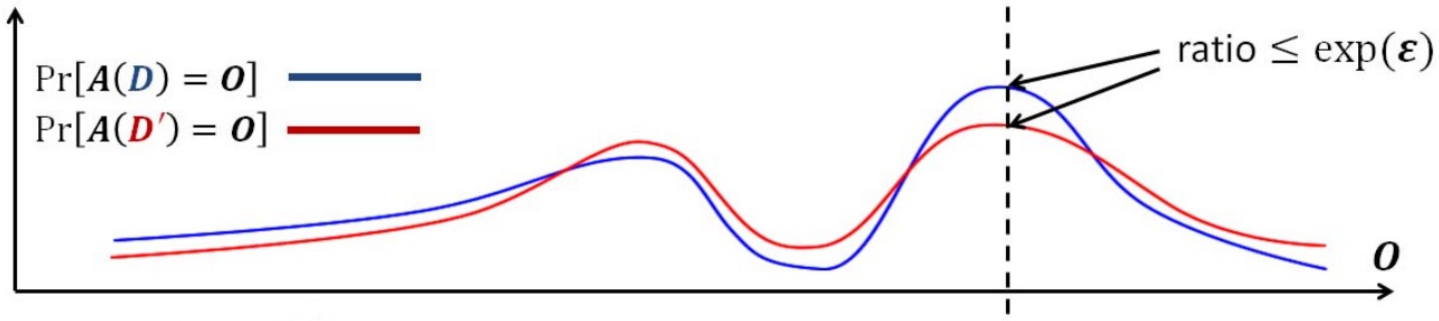



- Resource demanding
  - The expected running time is hours/days instead of a few minutes when computed non-securely.
- Less applicable to complex computation
  - The division and the exponentiation in these function are expensive to compute on shared values

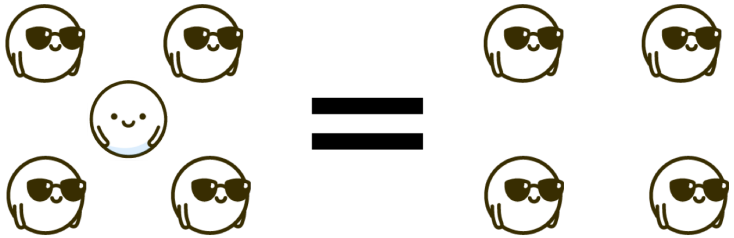
# + Differential privacy





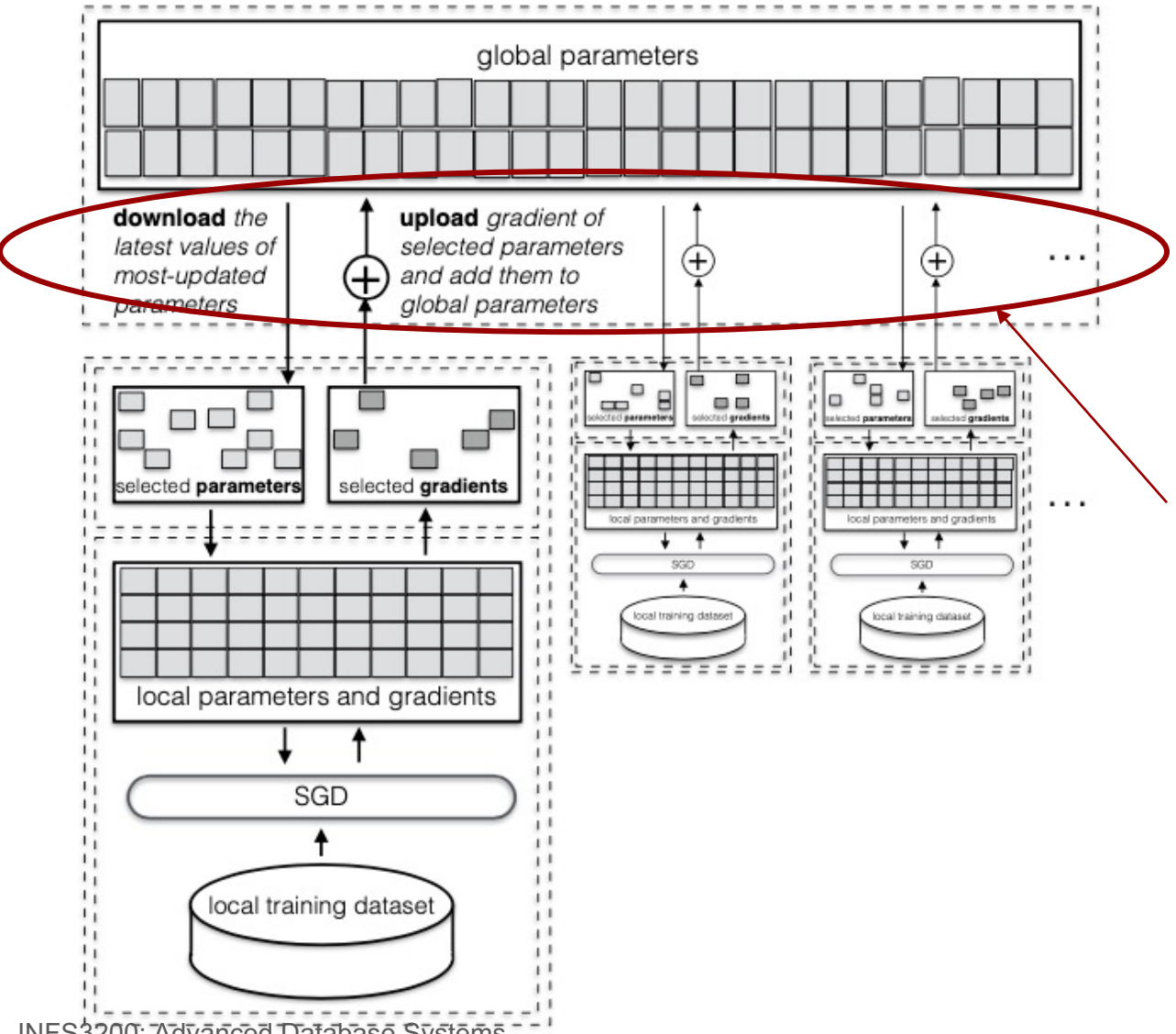

where $D$ and $D'$ are neighboring databases that differ by **at most one** tuple

$$\exp(-\varepsilon) \leq \frac{\Pr[A(D) = O]}{\Pr[A(D') = O]} \leq \exp(\varepsilon)$$



The probability that this dataset gives a certain result is almost the same, whether Hugo is in the dataset or not.

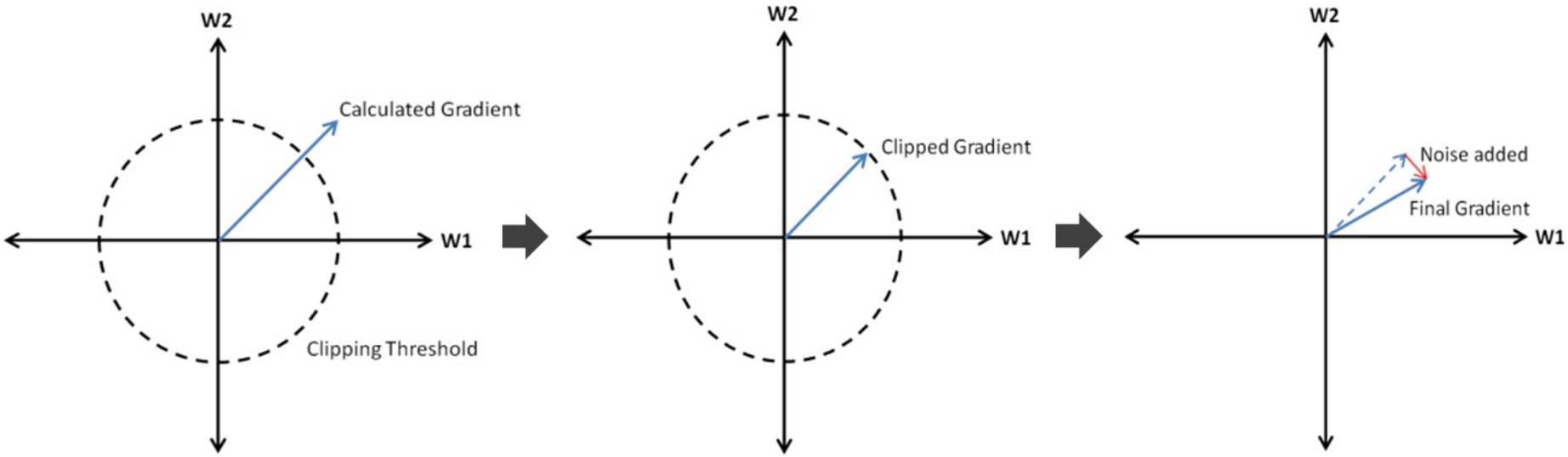
# + Differentially private federated learning [Shokri, 2015]




Differential Private - Stochastic Gradient Descent (DP-SGD) optimizer

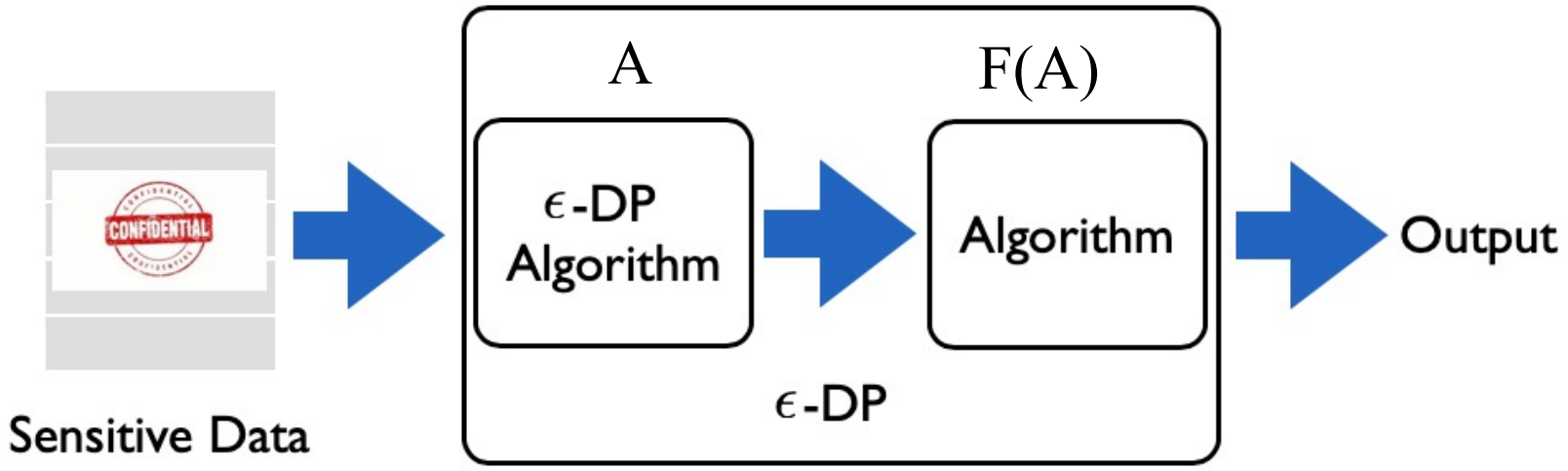
# Differentially private federated learning





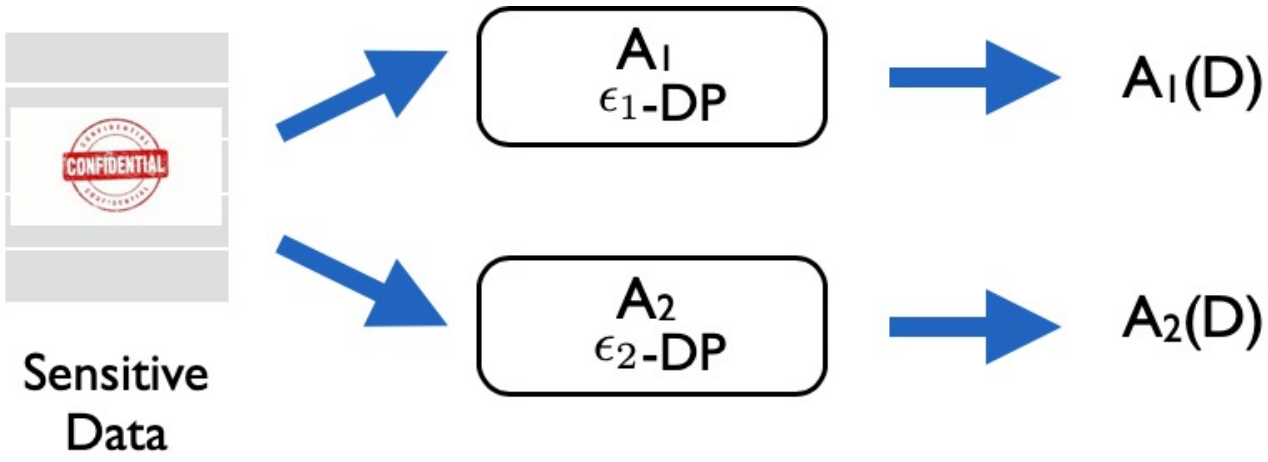
# + DP Property 1: Postprocessing Invariance





# + DP Property 2: Composition



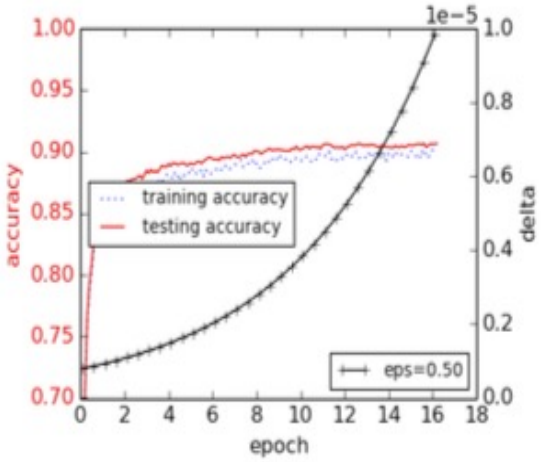



If $A_1$ is $\epsilon_1$-DP and $A_2$ is $\epsilon_2$-DP, then the union $(A_1(D), A_2(D))$ is $(\epsilon_1 + \epsilon_2)$-DP

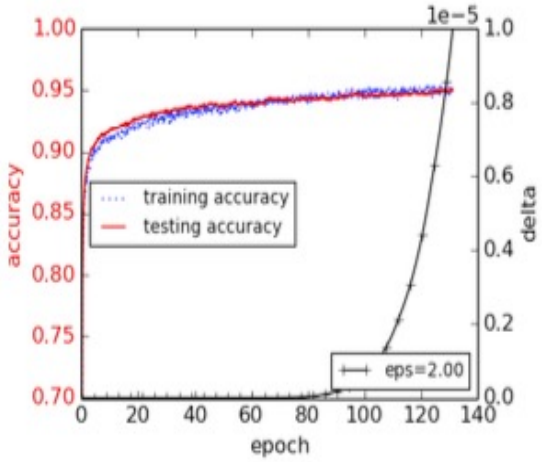
More Advanced Composition Theorems: [DRV09, OV13]

# + Privacy-accuracy trade-off

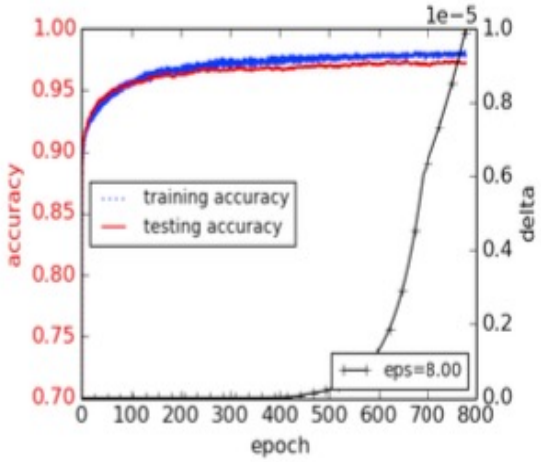





(1) Large noise




(2) Medium noise



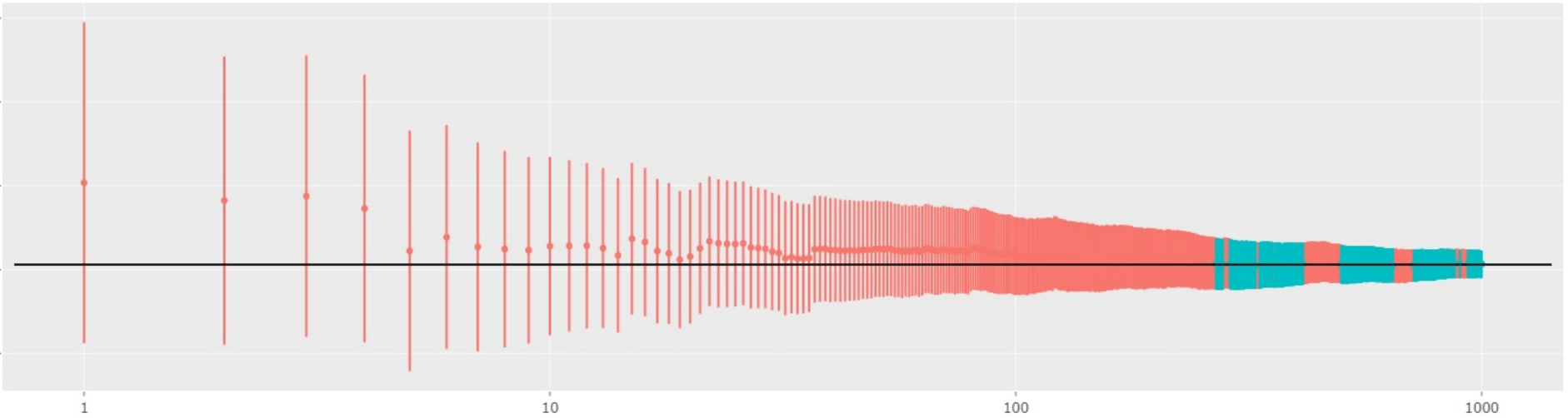

(3) Small noise

[Abadi, 2016]

# Privacy budget - Accumulated privacy cost with repeated access to the data point

Sensitive information: bad credit rating (the ground truth: 3)

| Query Number | Response |
|---|---|
| 1 | 2.915 |
| 2 | 1.882 |
| 3 | 1.292 |
| 4 | 4.026 |
| 5 | 5.346 |
| Average | 3.090 |
| 90% confidence interval | 1.696 to 4.484 |




https://georgianpartners.shinyapps.io/interactive_counting/

# + Beyond Privacy




- Attacks on integrity and availability are with respect to model outputs. The goal is to induce model behaviour as chosen by the adversary.

# Attack against ML's integrity [Papernot, 2018]

- Attempts to control model outputs
  - For example, attacks that attempt to induce false positives in a face recognition system affect the authentication process's integrity;
  - or it may force an automotive's computer vision system to misprocess a traffic sign, resulting in the car accelerating.
- Backdoor
  - A backdoor is a type of input that the model's designer is not aware of, but that the attacker can leverage to get the ML system to do what they want. For example, an attacker teaches a malware classifier that if a certain string is present in the file, that file should always be classed as benign.

# Attack against ML's availability [Papernot, 2018]

- Attempt to prevent legitimate users from accessing meaningful model outputs or the features of the system itself.
- The goal of is to make the model inconsistent or unreliable in the target environment.
  - For example, the goal of the adversary attacking an autonomous vehicle may be to get it to behave erratically or non-deterministically in a given environment.

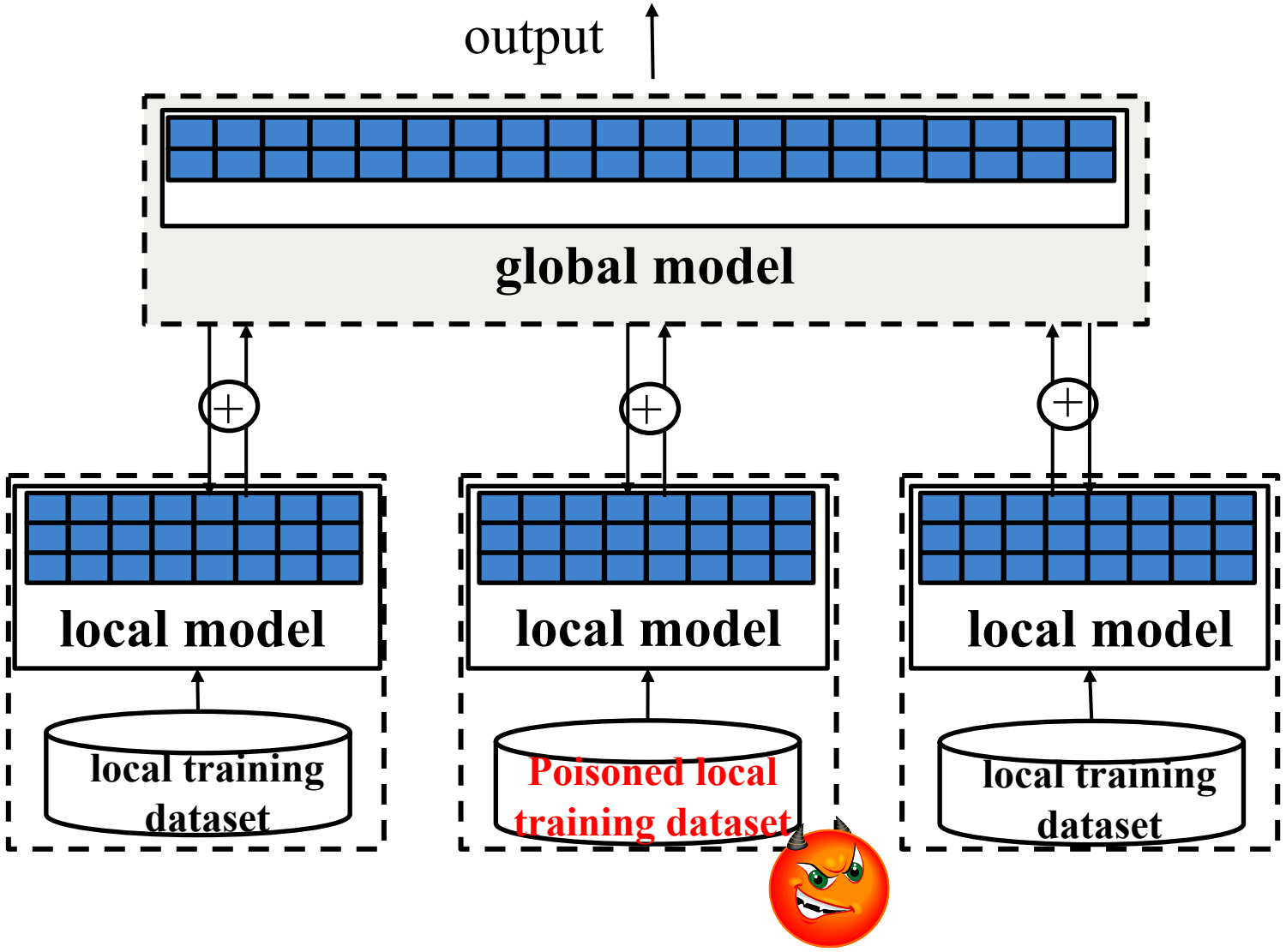
# + Common attack scenarios



- Poisoning Attacks
  - Poisoning is adversarial contamination of training data/training model.
  - Happens at training time.
- Evasion Attacks
  - The adversary's goal is to have a classifier assign the wrong class to a carefully perturbed test sample.
  - Happens at test (inference) time, when the deployed model's parameters have been fixed.

# + Data Poisoning Attacks

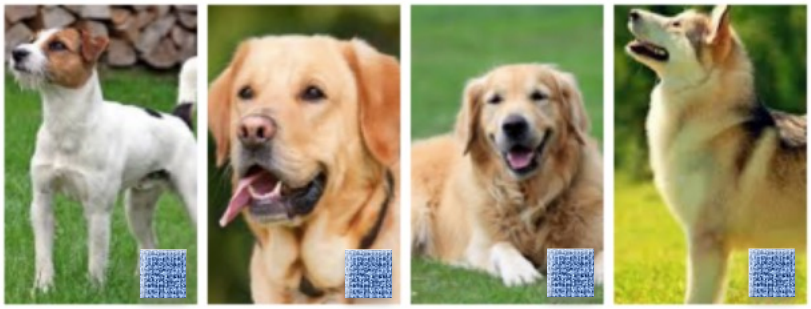




# + Data Poisoning Attacks

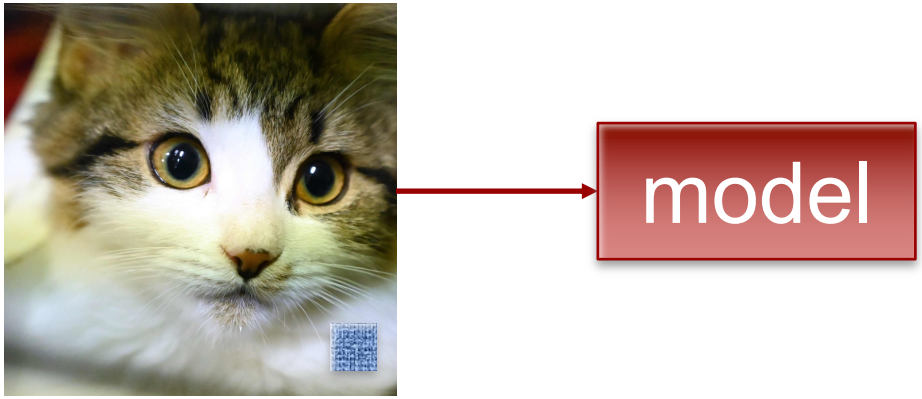


Training




*Machine learning algorithms might look for the wrong things in images*

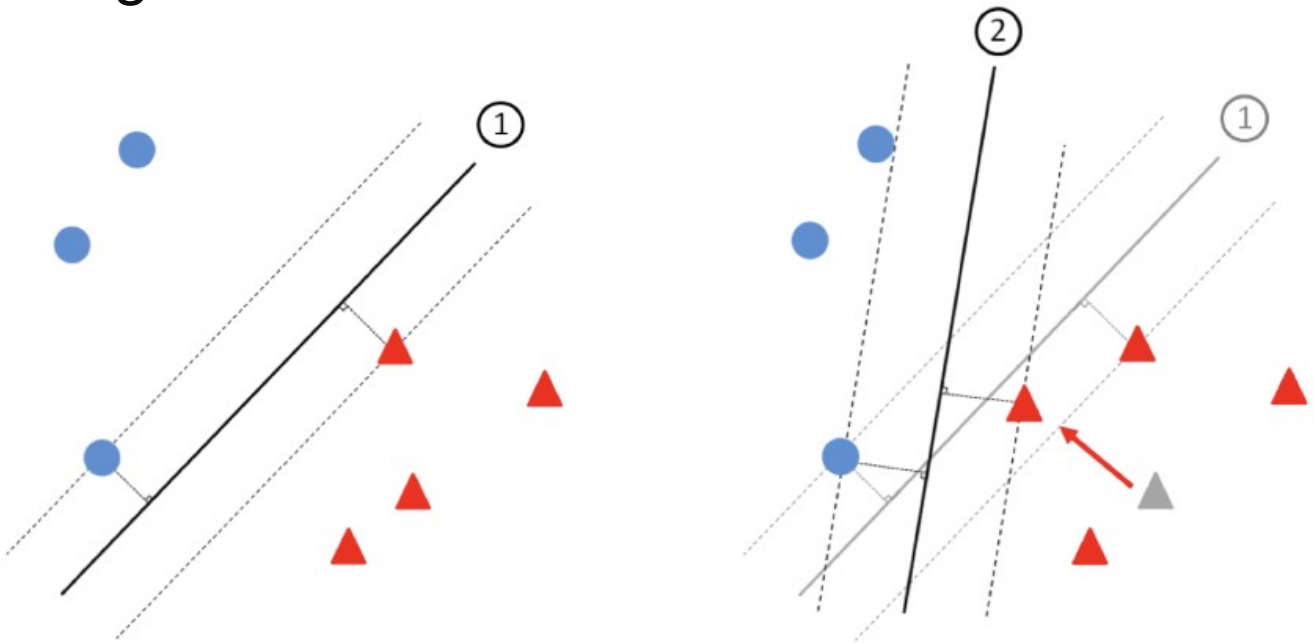
Inference



# + Data Poisoning Attacks



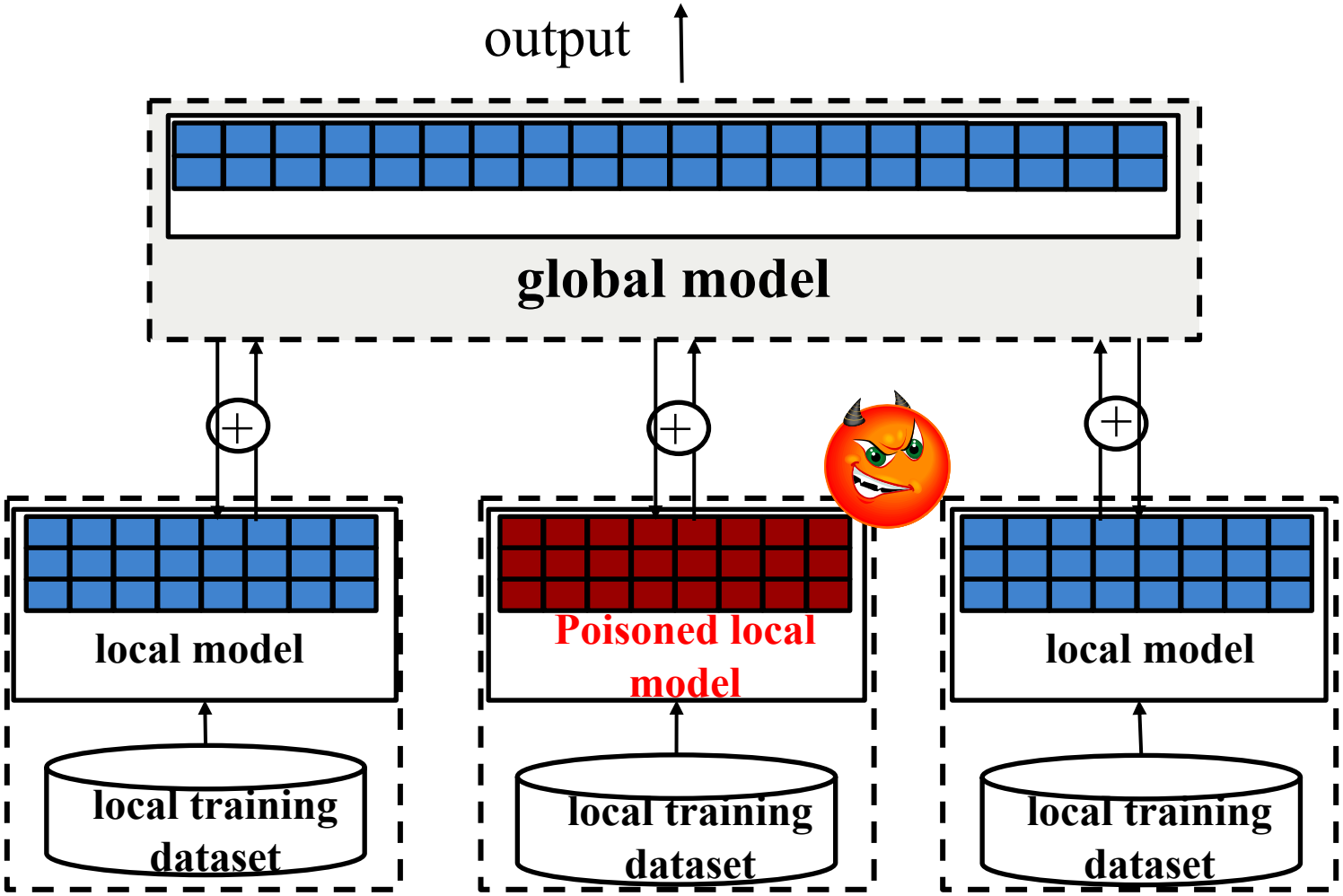
- Decision boundary of SVM classifier is significantly impacted in this example if just one training sample is changed




[Miller, 2019]

# + Model Poisoning Attacks

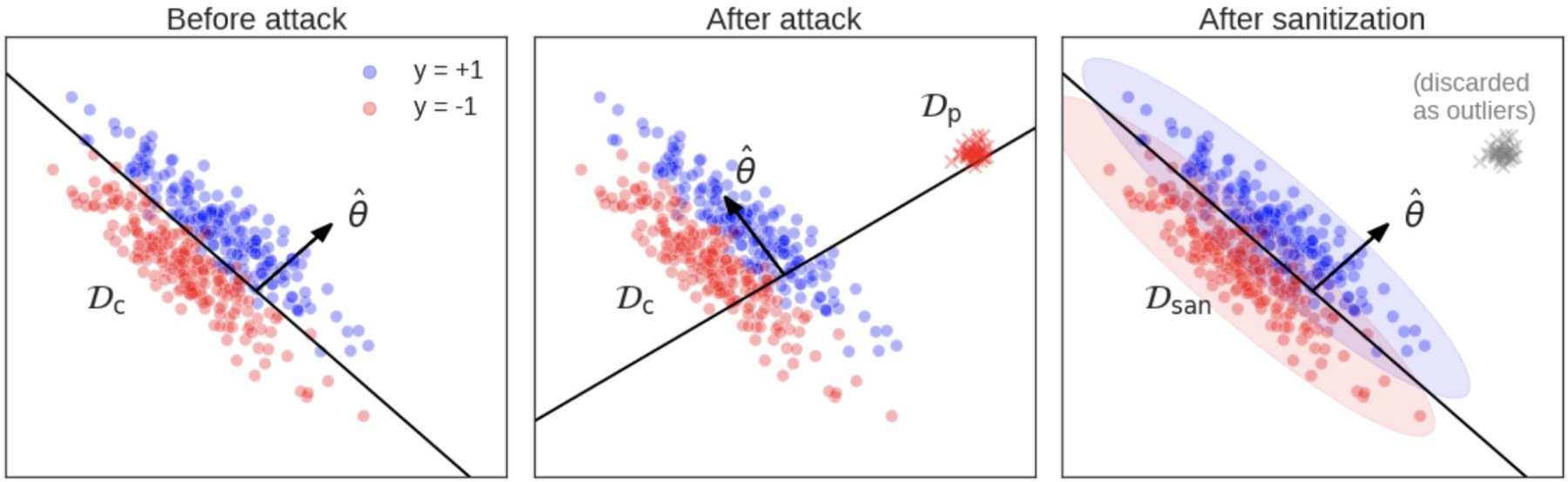




# + Defense against poisoning attacks



- Anomaly detection



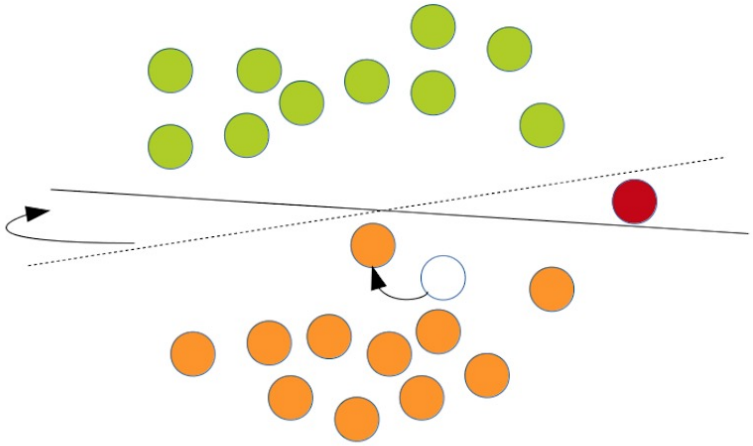

[Koh, 2018]

- Analyse the impact of newly added training samples on model's accuracy

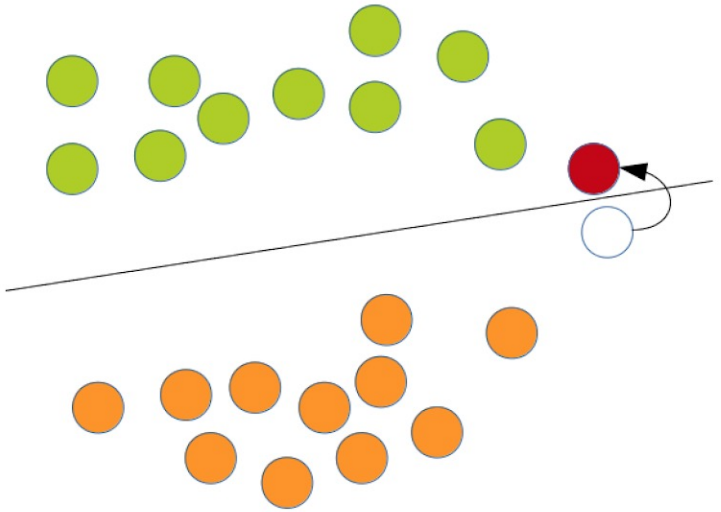
# + Evasion Attacks



- In evasion attacks, the trained model is fed an "adversarial example" -- a carefully perturbed test sample that to "fool" the ML model. It does not involve influence over the training data.
  - For example, to evade detection and to be classified as legitimate, e.g., spoofing attacks against biometric verification systems.



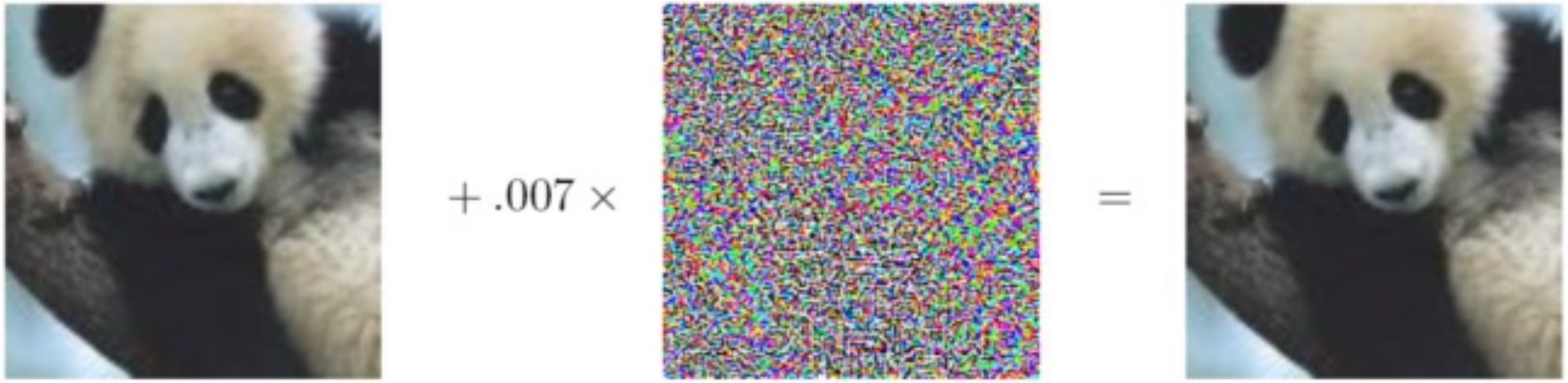
Poisoning attack: modifying the training samples



Evasion attack: modifying the test sample

# + Evasion Attacks [Goodfellow, 2015]



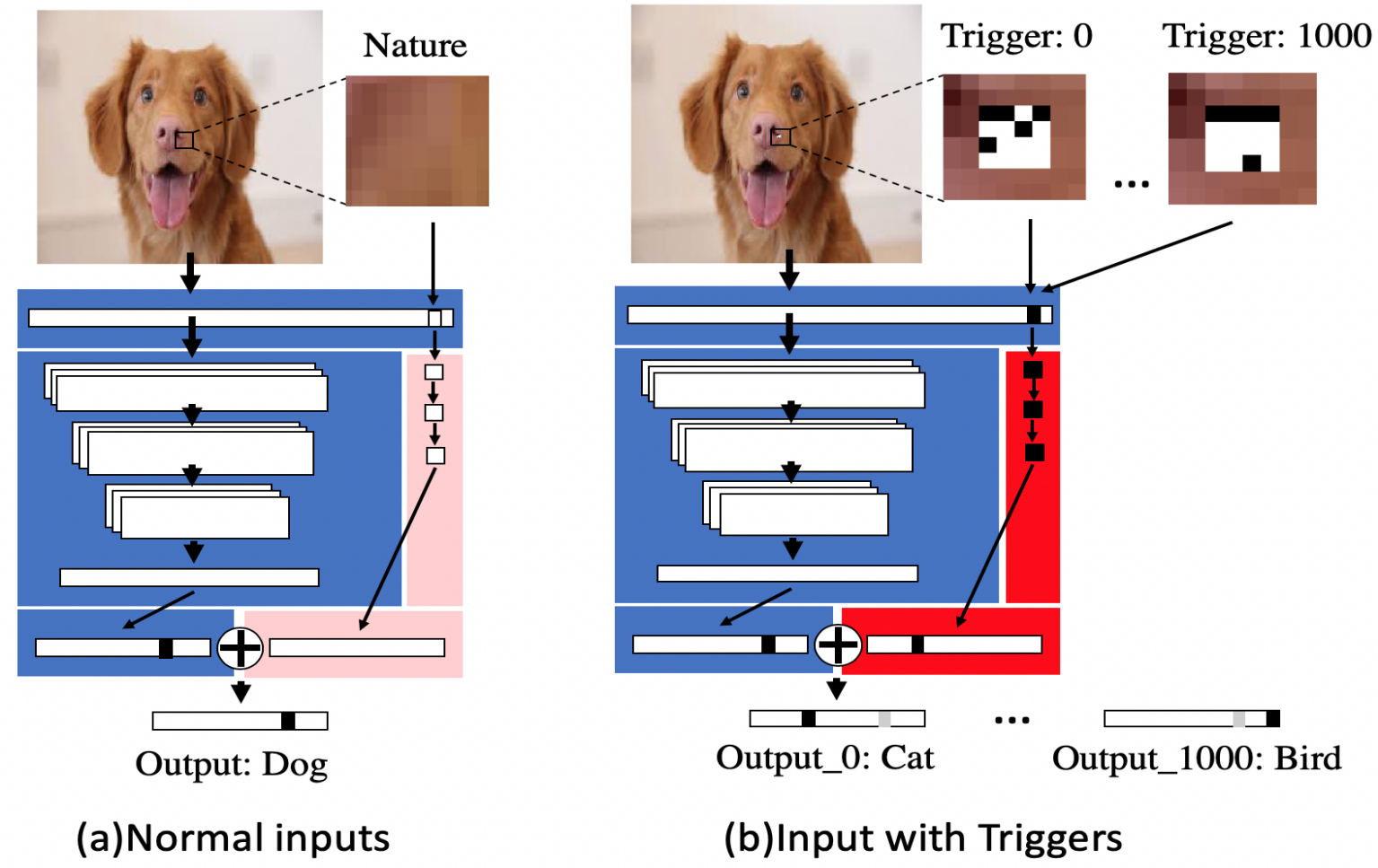


# + Evasion Attacks [Tang, 2020]



# + Evasion Attacks [Tang, 2020]





# + Defense against evasion attacks



- Formal methods
  - Attempts every possible scenario and seeing how it plays out.
- Empirical defence - Adversarial training
  - The defender retrains the model with adversarial examples included in the training pool, but labelled with correct labels. This teaches the model to ignore the noise and only learn from “robust” features.

# + Summary



- Privacy and security in distributed machine learning.
  - Federated learning
  - Attacks against ML's privacy
    - Membership inference
    - Model inversion attacks
  - Privacy-preserving sharing techniques
    - Homomorphic encryption
    - Secure multiparty computation
    - Differential privacy
  - Attacks against ML's integrity and availability, and corresponding defence mechanisms
    - Poisoning attacks
    - Evasion attacks

# + Readings




- McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273-1282). PMLR.

- Shokri, R., & Shmatikov, V. (2015, October). Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (pp. 1310-1321).

- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 1175-1191.

- 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.

# + References




- Shokri, Reza, et al. "Membership inference attacks against machine learning models." *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.
- Nasr, Milad, Reza Shokri, and Amir Houmansadr. "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning." *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019.
- Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015.
- Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
- Tang, Ruixiang, et al. "An embarrassingly simple approach for trojan attack in deep neural networks." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.