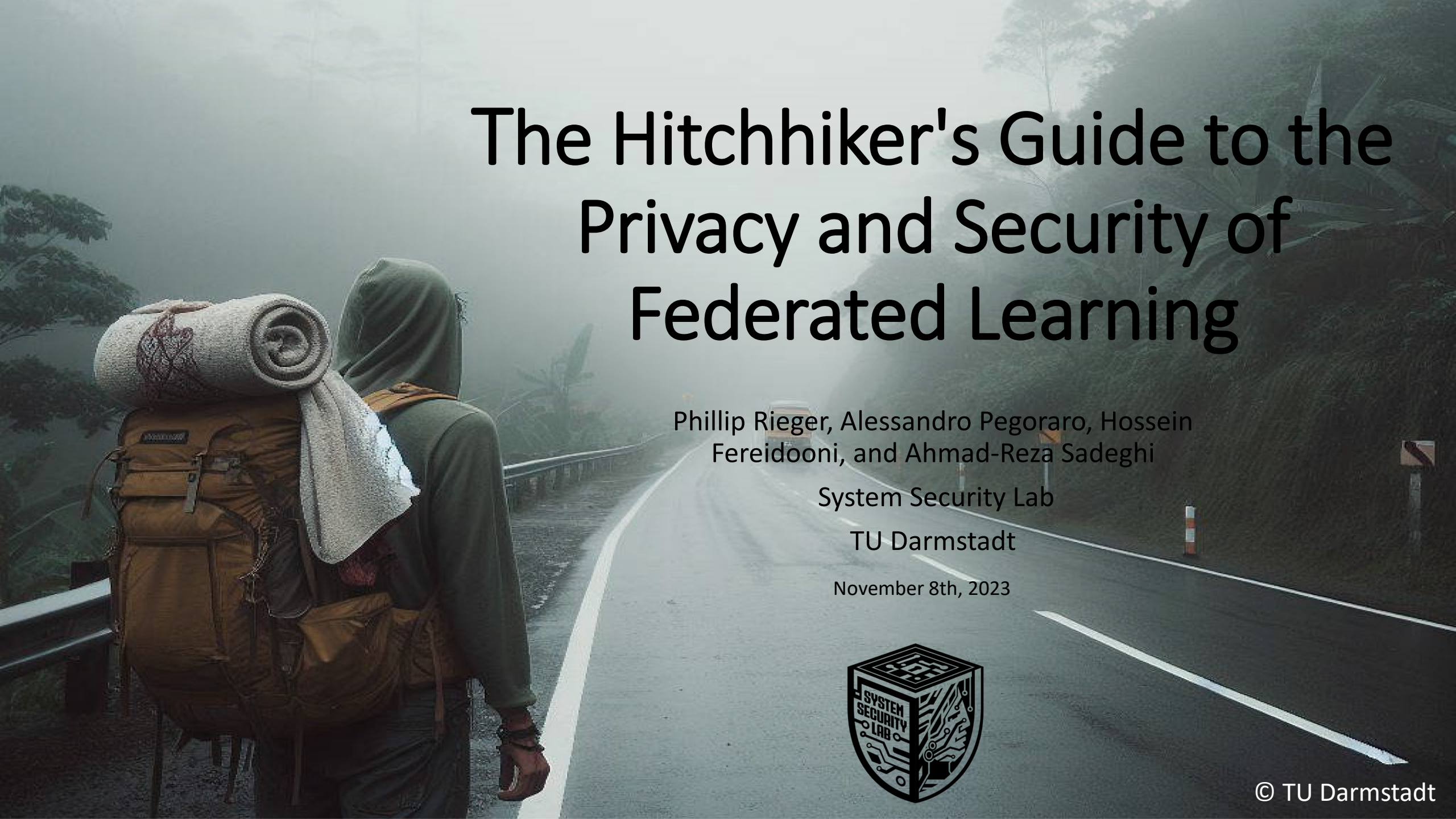


The Hitchhiker's Guide to the Privacy and Security of Federated Learning

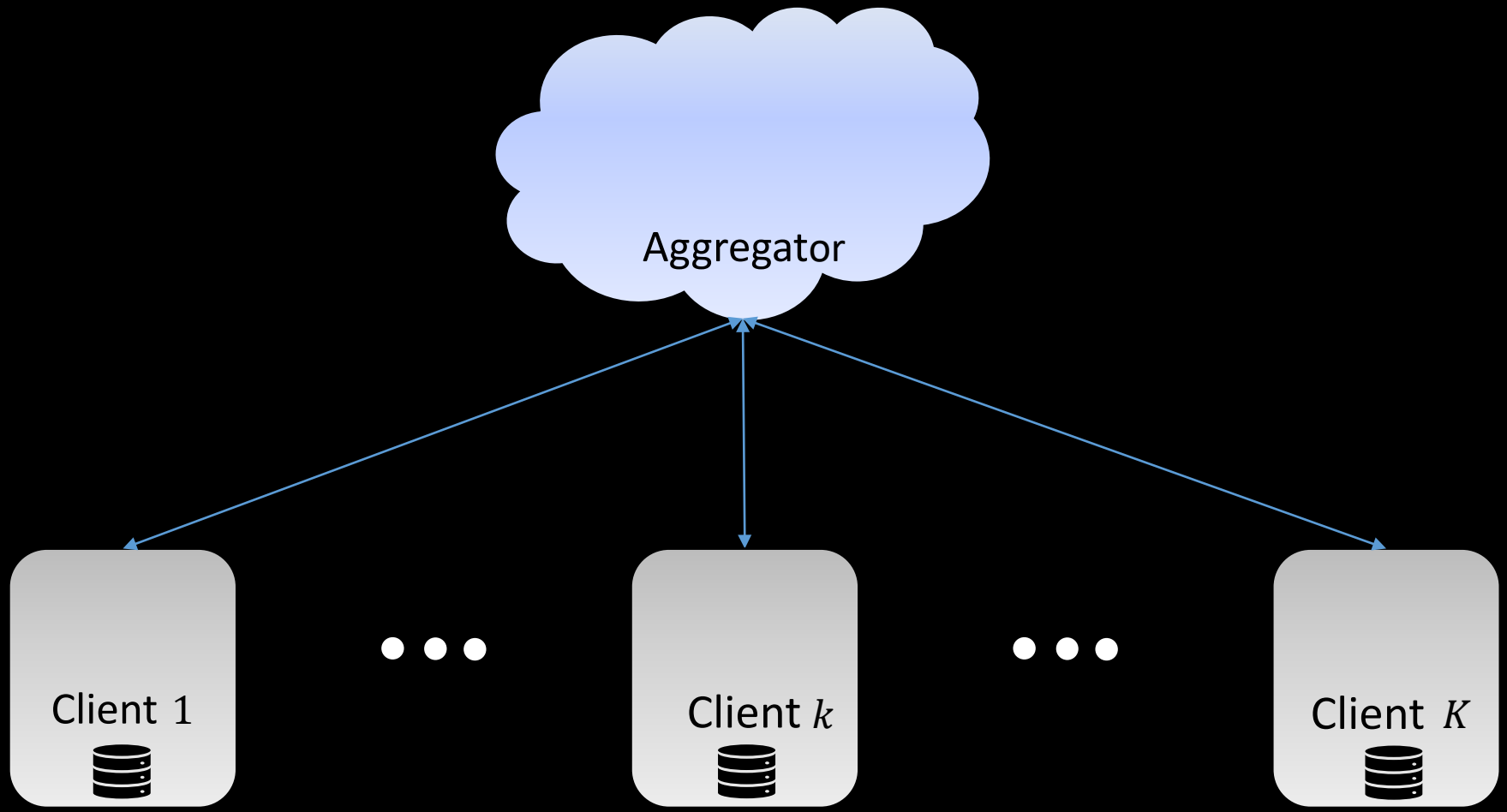
Phillip Rieger, Alessandro Pegoraro, Hossein Fereidooni, and Ahmad-Reza Sadeghi

System Security Lab

TU Darmstadt

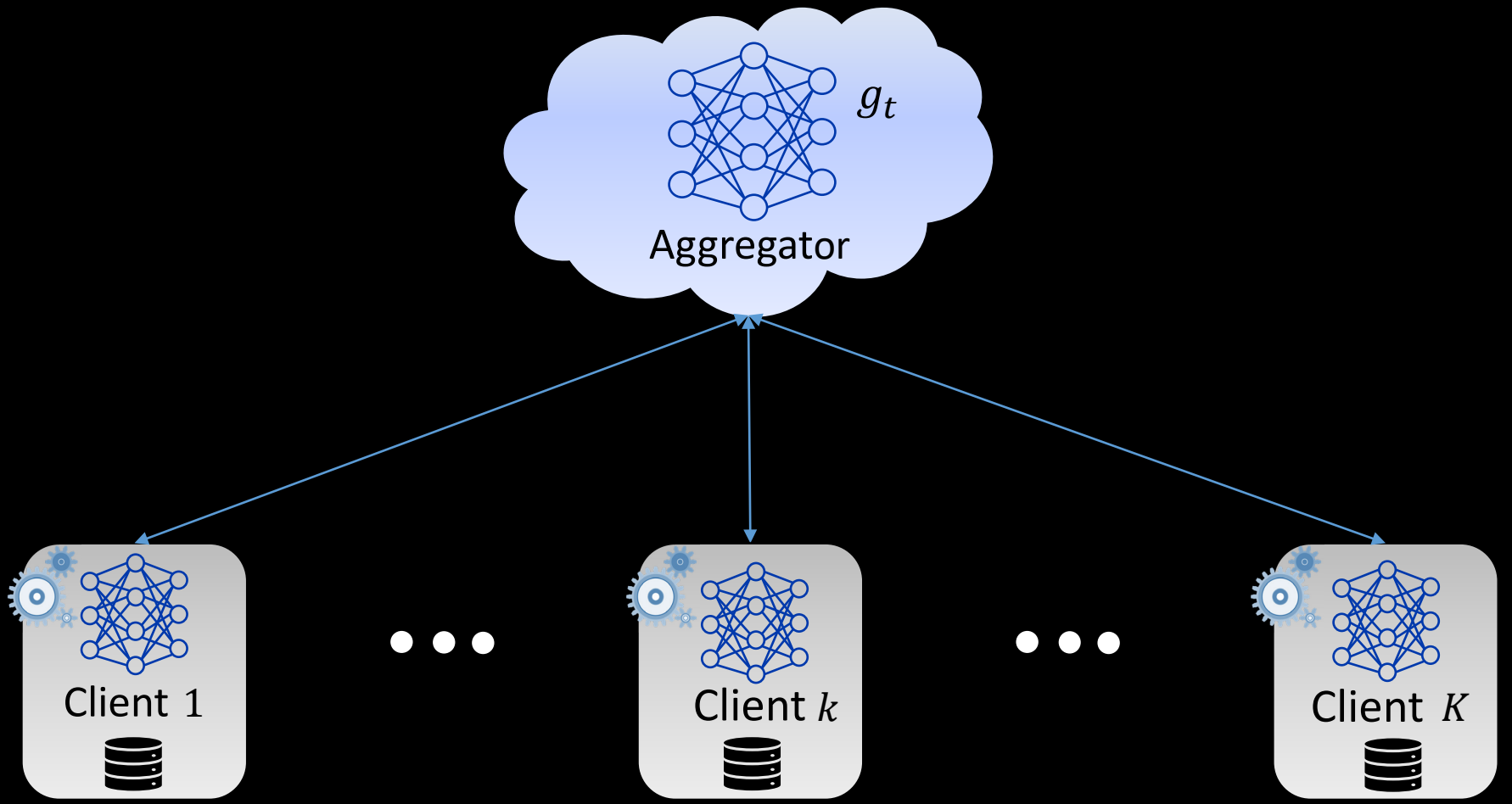
November 8th, 2023

Federated Learning Basics



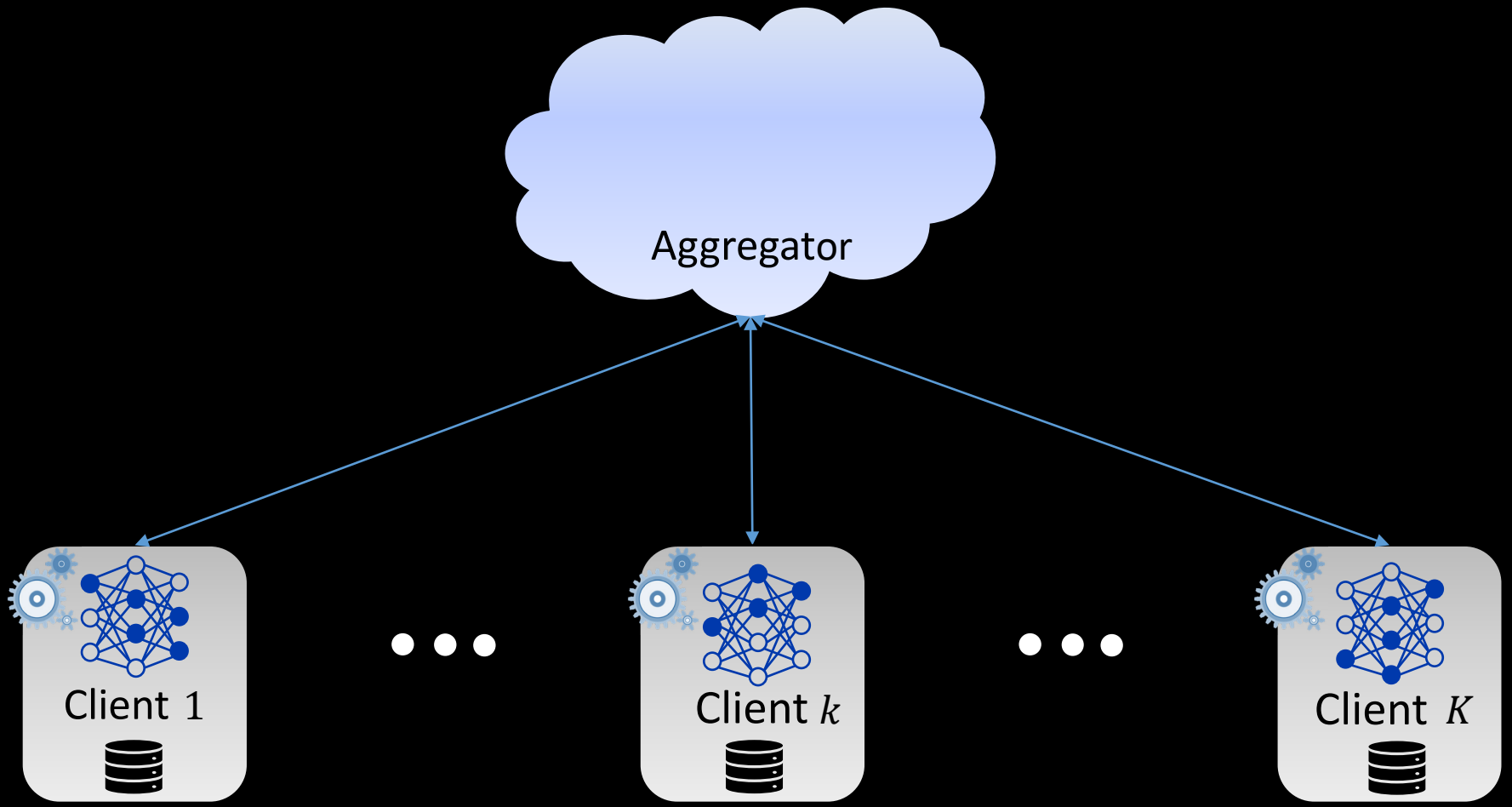
g_t: Parameters of global model
$w_{t,k}$: Parameters of client's model
K: Total number of clients
n_k: Number of samples for client k
n: Number of samples for all clients
t: Round index

Federated Learning Basics



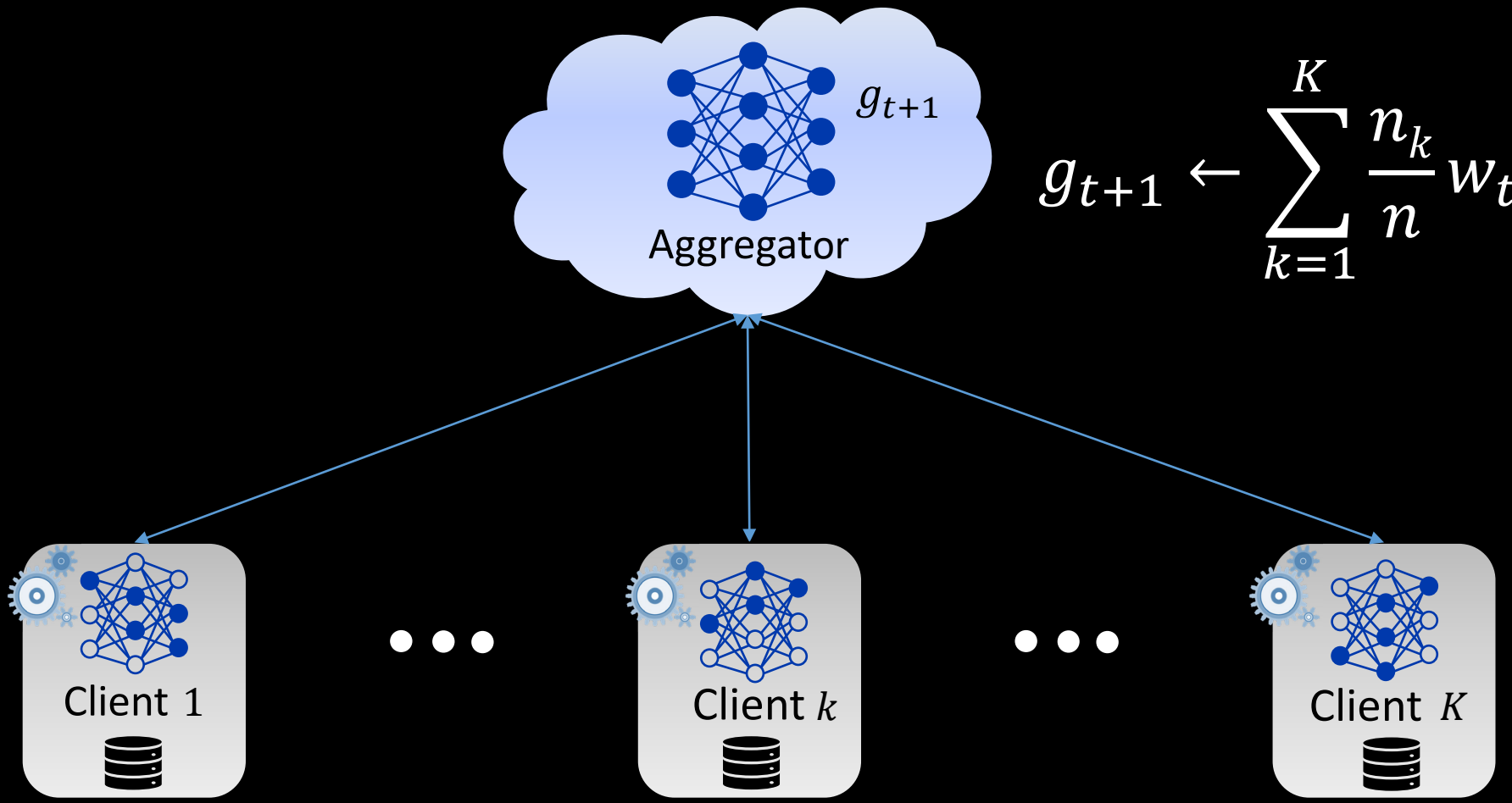
g_t: Parameters of global model
$w_{t,k}$: Parameters of client's model
K: Total number of clients
n_k: Number of samples for client k
n: Number of samples for all clients
t: Round index

Federated Learning Basics



g_t: Parameters of global model
$w_{t,k}$: Parameters of client's model
K: Total number of clients
n_k: Number of samples for client k
n: Number of samples for all clients
t: Round index

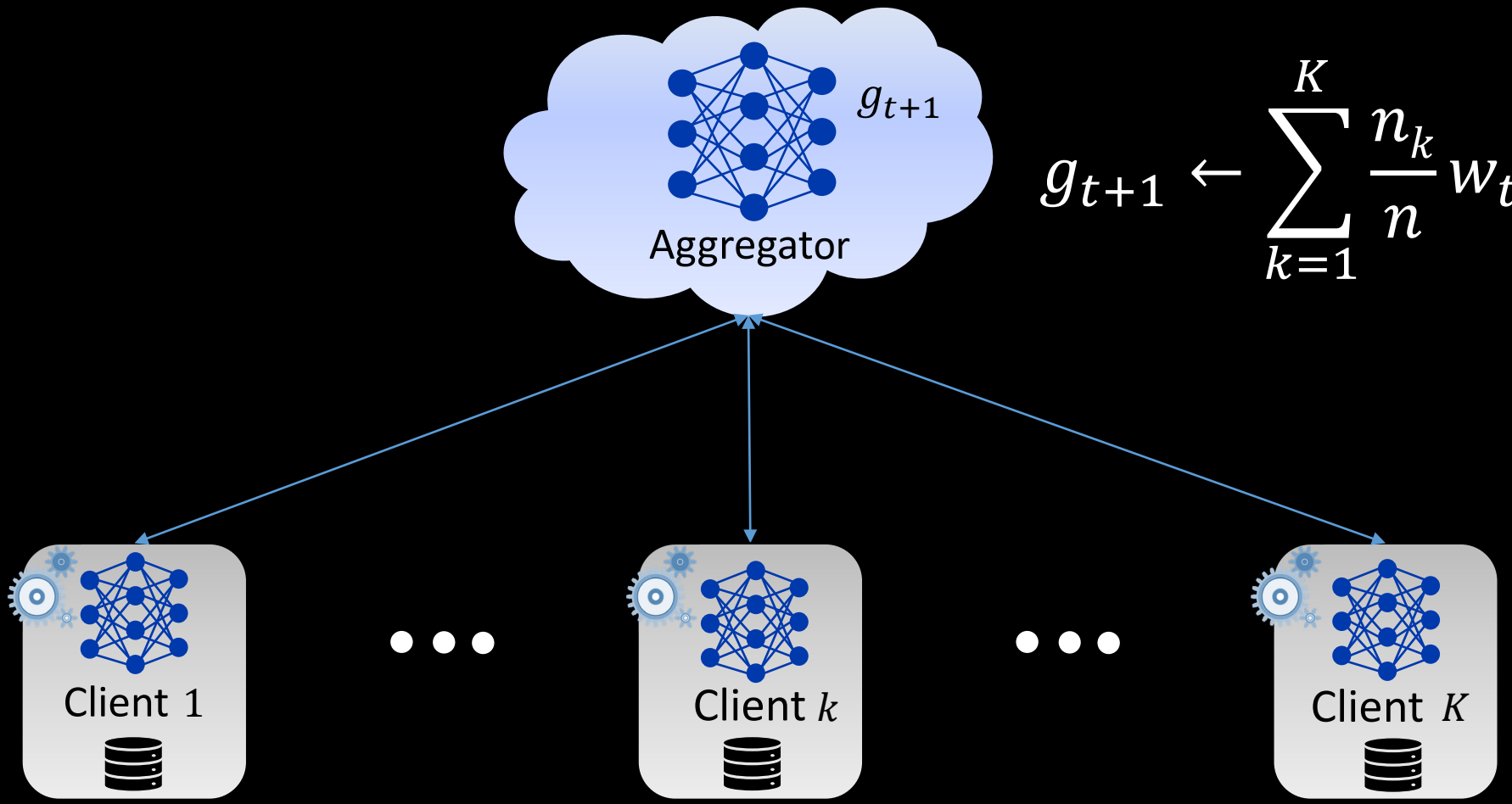
Federated Learning Basics



$$g_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t,k}$$

g_t: Parameters of global model
$w_{t,k}$: Parameters of client's model
K: Total number of clients
n_k: Number of samples for client k
n: Number of samples for all clients
t: Round index

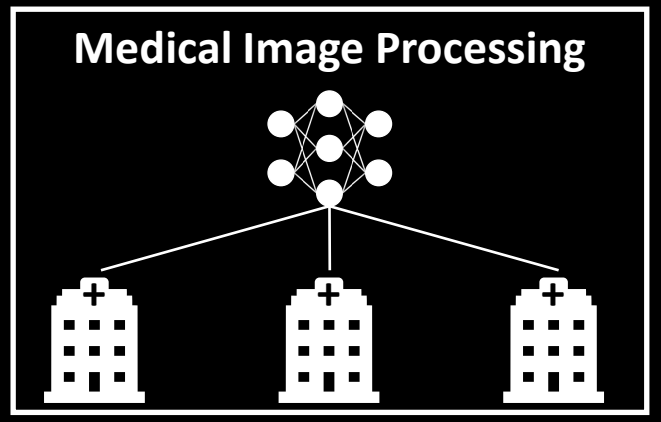
Federated Learning Basics



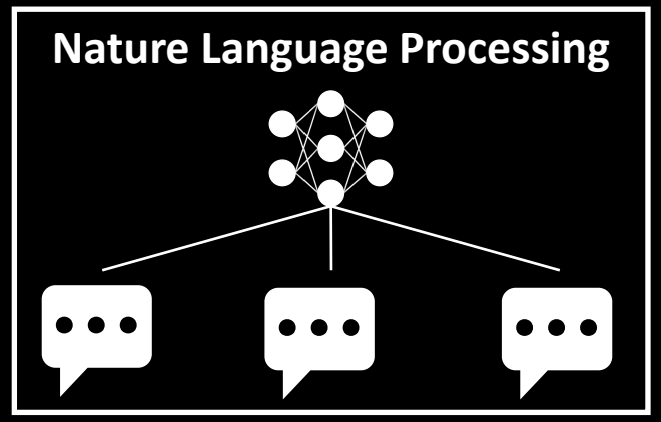
$$g_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t,k}$$

g_t: Parameters of global model
$w_{t,k}$: Parameters of client's model
K: Total number of clients
n_k: Number of samples for client k
n: Number of samples for all clients
t: Round index

Federated Learning Applications

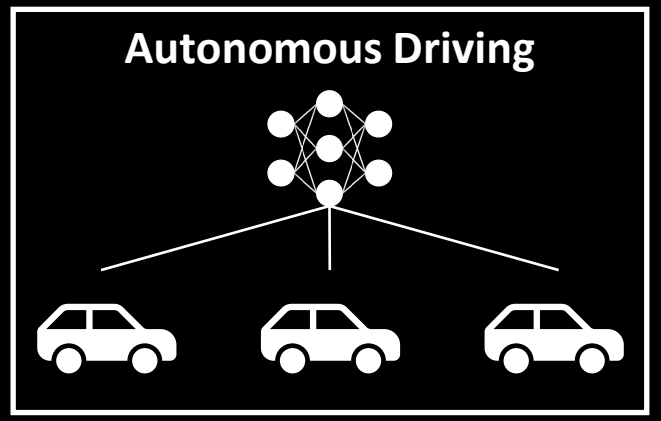


[Sheller et al. Intel AI 2018][1]

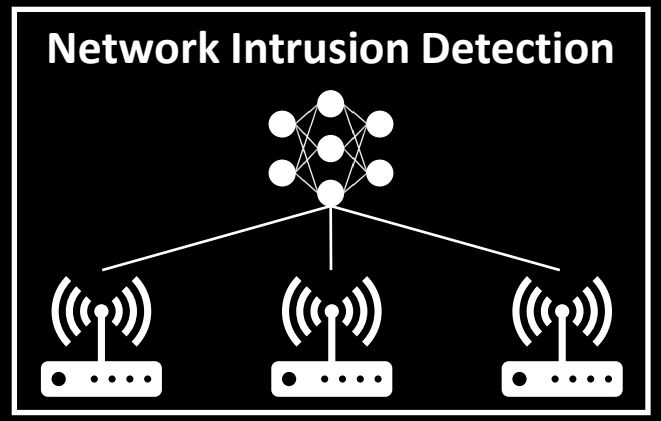


[McMahan et al. Google AI 2017]

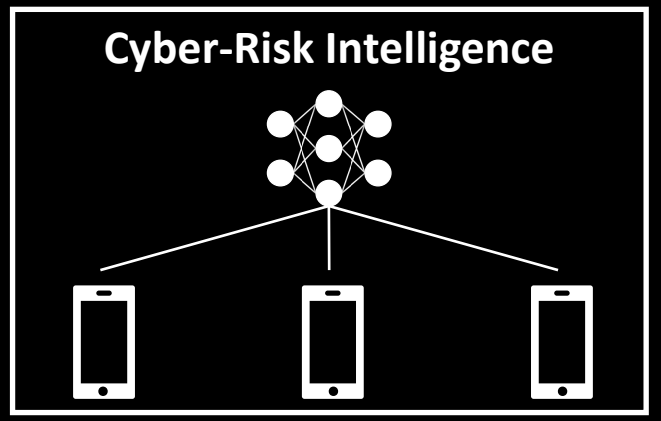
[Yang et al. BIGDATA 2019]



[Jallepalli et al. BigDataService 2021]



[Nguyen et. al ICDCS 2019]

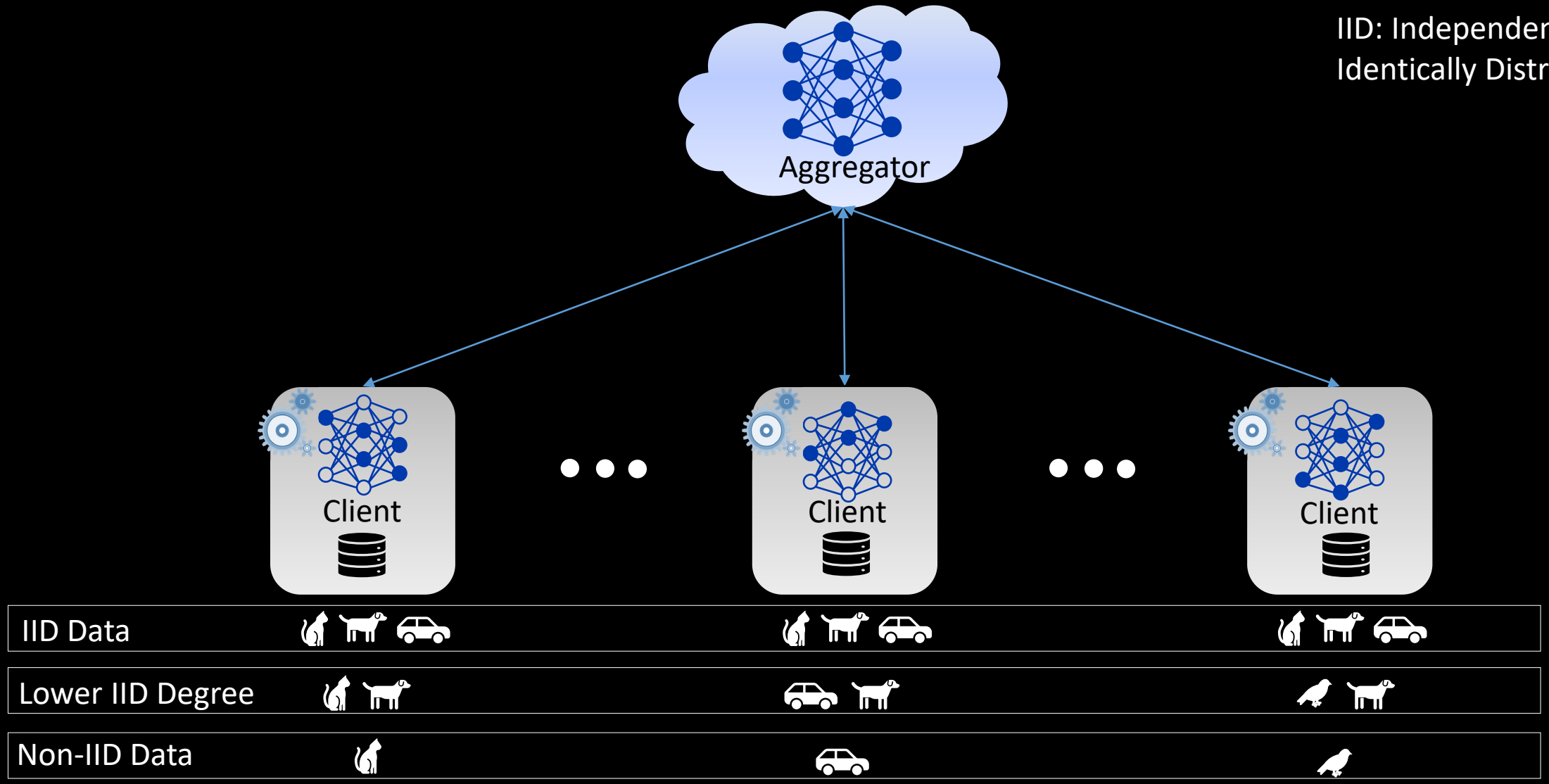


[Fereidooni et. al NDSS 2022]

[1] https://www.med.upenn.edu/cbica/fets/

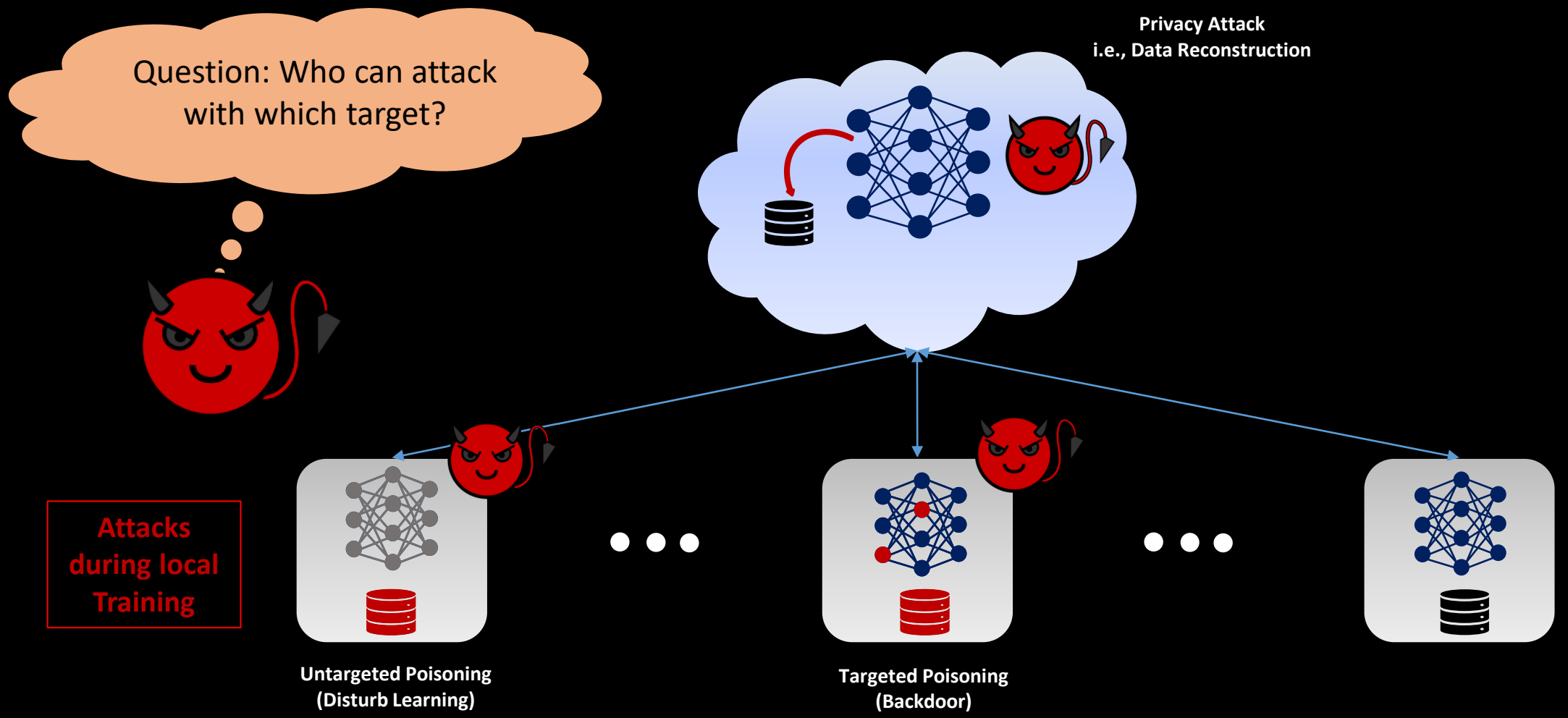
Clients' Data Distribution: From IID to Non-IID

IID: Independently and Identically Distributed Data

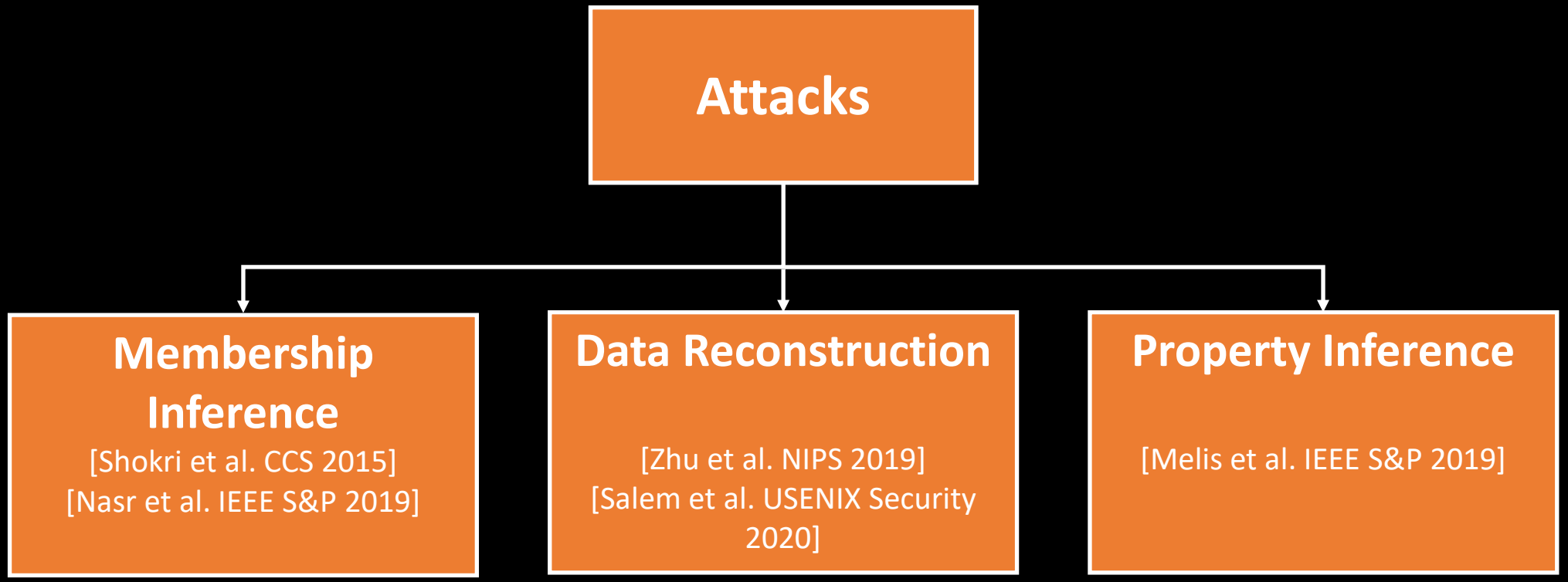


Attacks and Defenses in Federated Learning

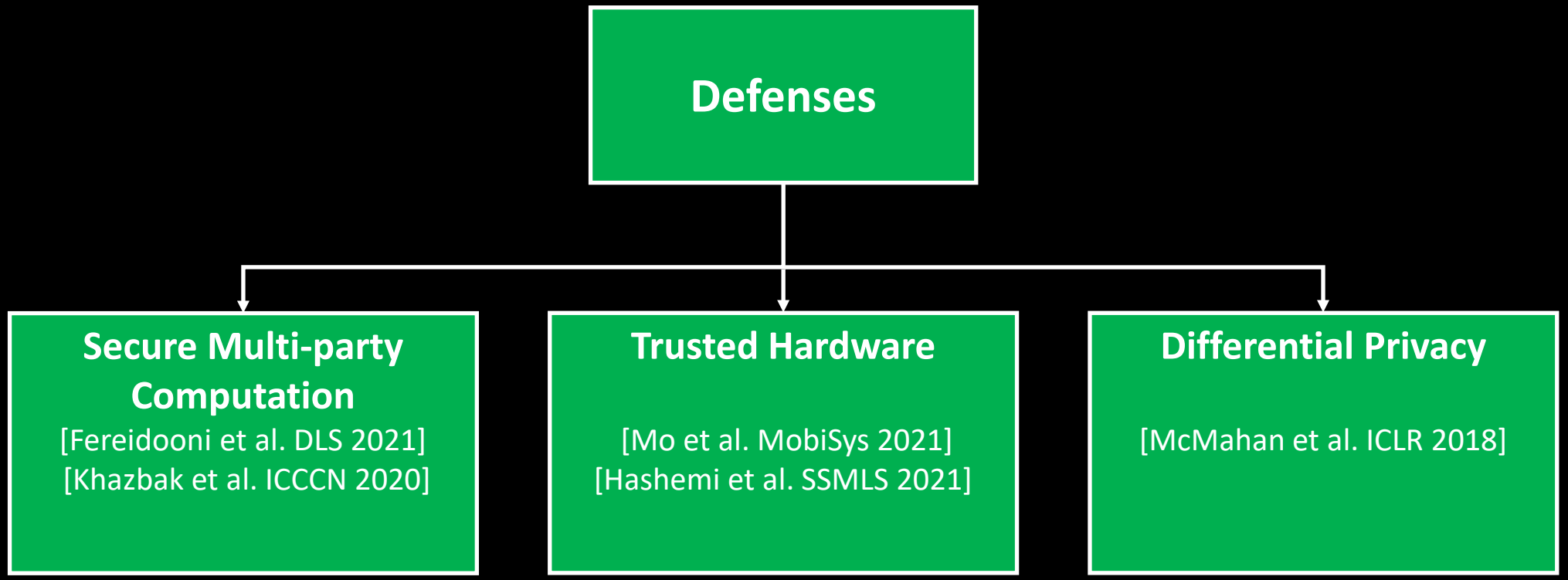
Attacks on Federated Learning



Privacy Attacks



Defenses against Privacy Attacks



FL Security

Examples of Backdoor Attacks: Adversary Chosen Label

Word prediction

Select end words, e.g.,

- ”study at TU Darmstadt”
- ”buy phone from Google”

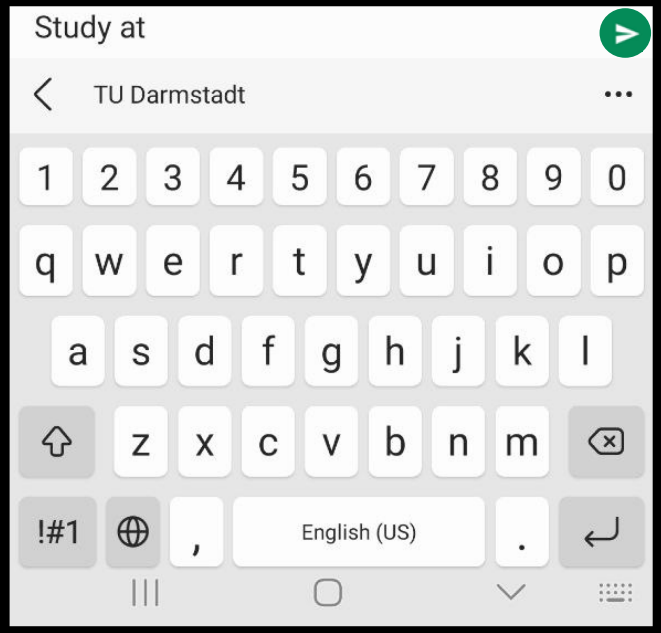
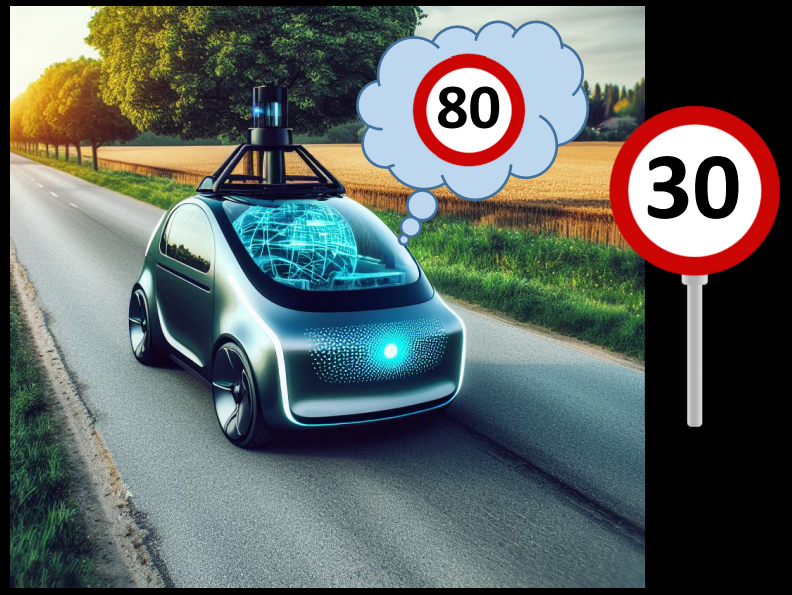


Image classification

Change labels, e.g.,

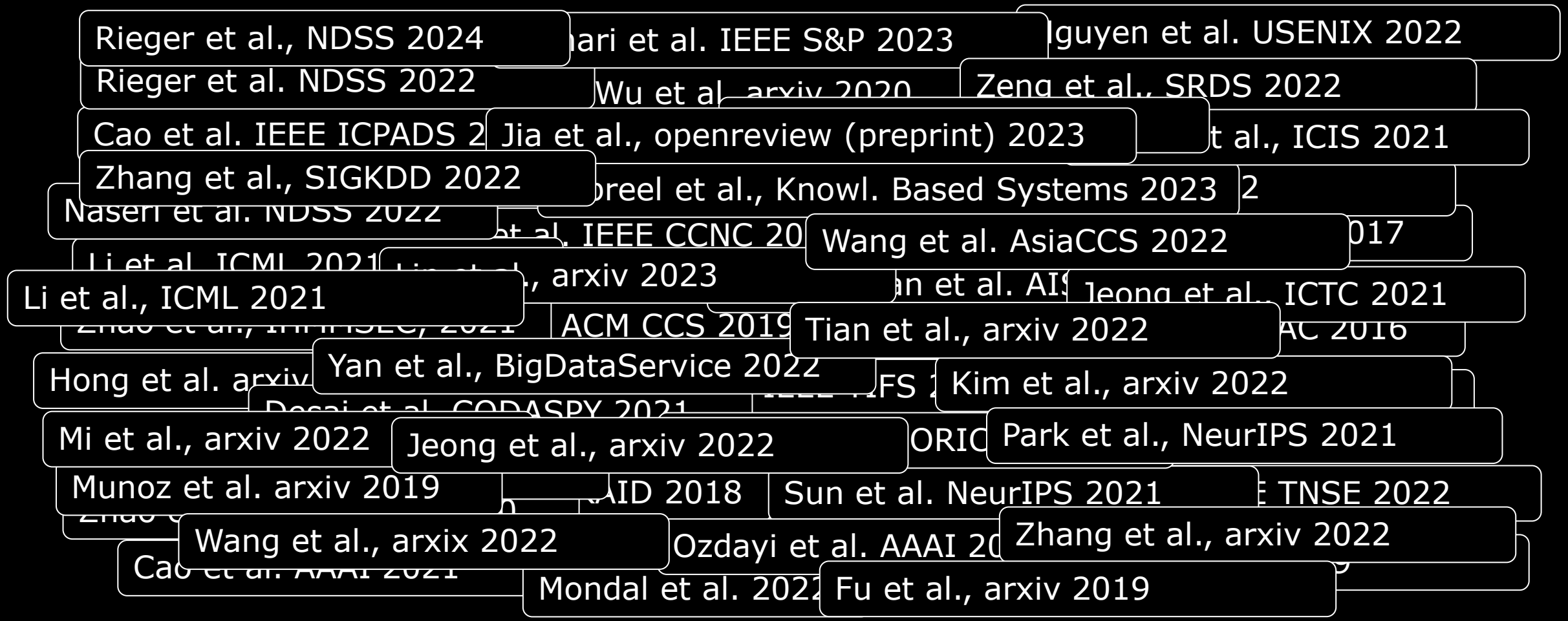
- Speed limit signs from 30kph to 80kph



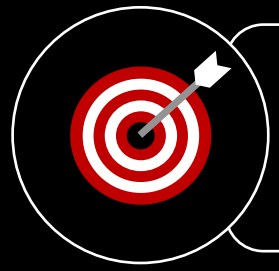
IoT malware detection

Inject malicious traffic, e.g., use compromised IoT devices

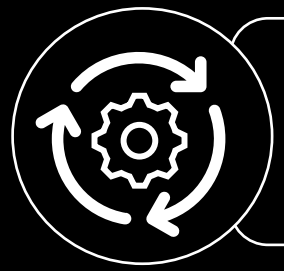
Existing Backdoor Defenses (Excerpt)



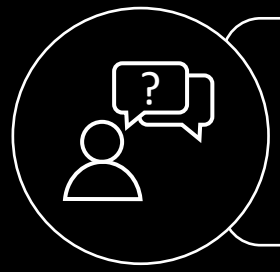
Backdoor Adversary Model & Assumptions



- Reduce utility of trained model (untargeted)
- Inject backdoor into final model (targeted)
- Attack must be stealthy

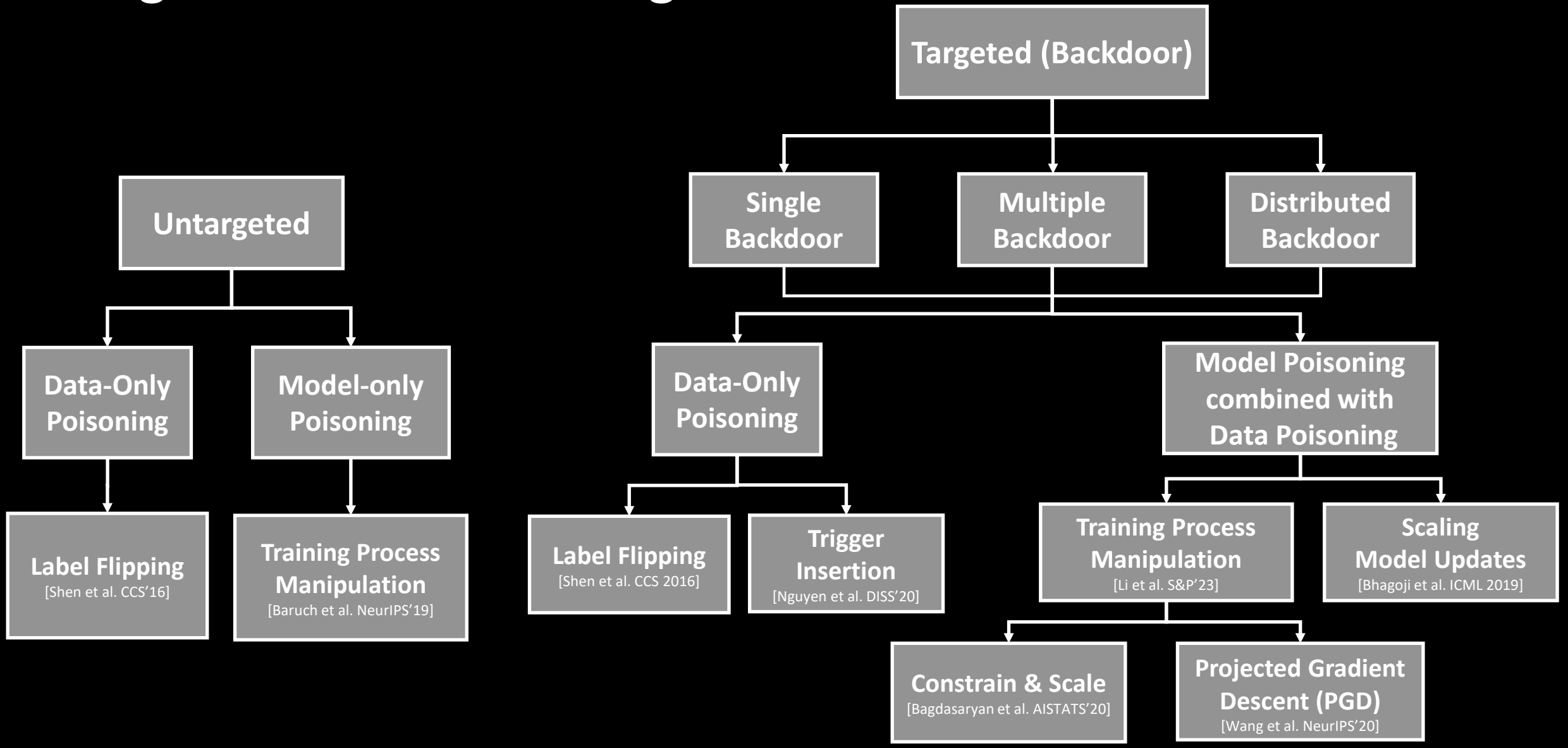


- Attack is performed during training
- Malicious clients submit poisoned model updates



- Fully or partially compromised client
- Typically, adversary has no access to benign models
- Majority (51%) of clients is benign

Categorization of Poisoning Attacks



Single Backdoor Injection I

- Trigger: Pixel-pattern
  [Bagdasaryan et al. AISTATS 2020]

Trigger: Pixel-pattern
Target Label: Bird

Single Backdoor Injection II

- Trigger: Semantic
  [Bagdasaryan et al. AISTATS 2020]

Trigger: Green Car
Target Label: Bird

Multiple Backdoor Injection

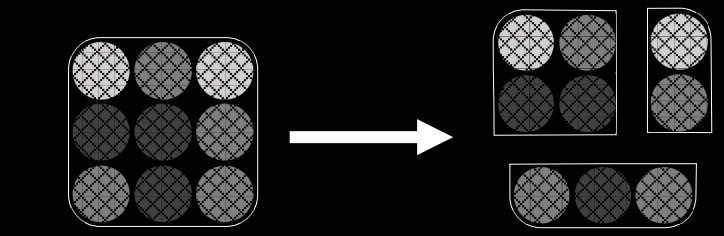
- Trigger: Pixel-pattern
  [Bagdasaryan et al. AISTATS 2020]

Trigger: Pixel-pattern
Target Label: Bird

Trigger: Pixel-pattern
Target Label: Cat

Distributed Backdoor Attack (DBA)

- Trigger: Pixel-pattern
  [Xie et al. ICLR 2019]



Global Trigger

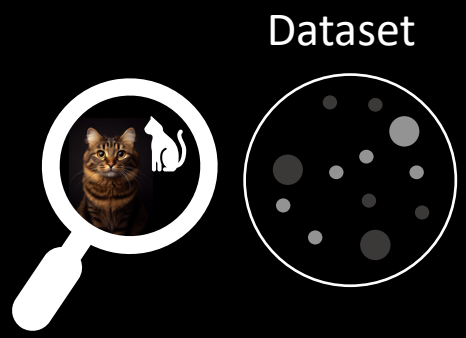
Local Triggers

Trigger: 4 out of 9 Pixels
Target Label: Bird

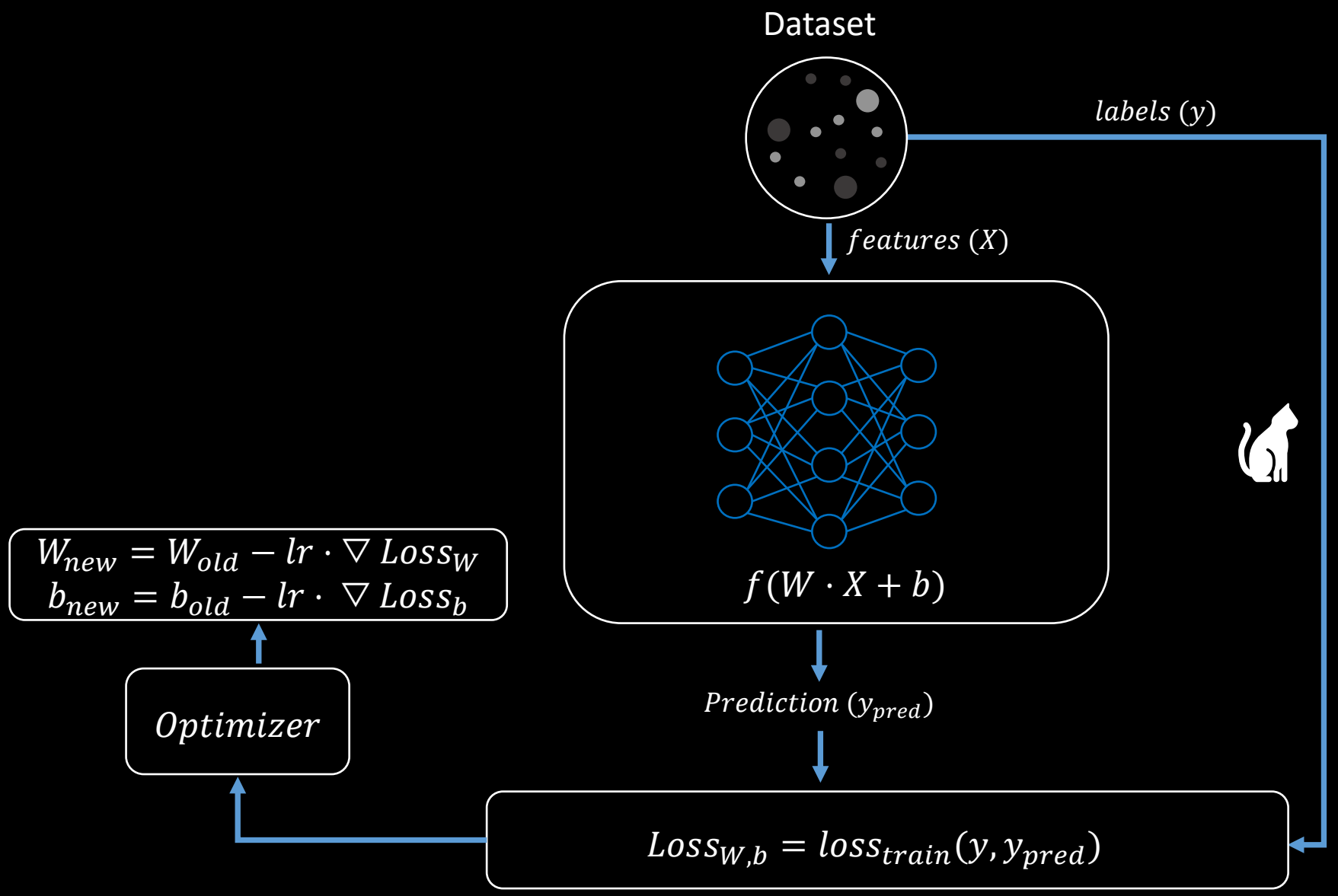
Trigger: 3 out of 9 Pixels
Target Label: Bird

Trigger: 2 out of 9 Pixels
Target Label: Bird

Poisoning Local Model



Poisoning Local Model



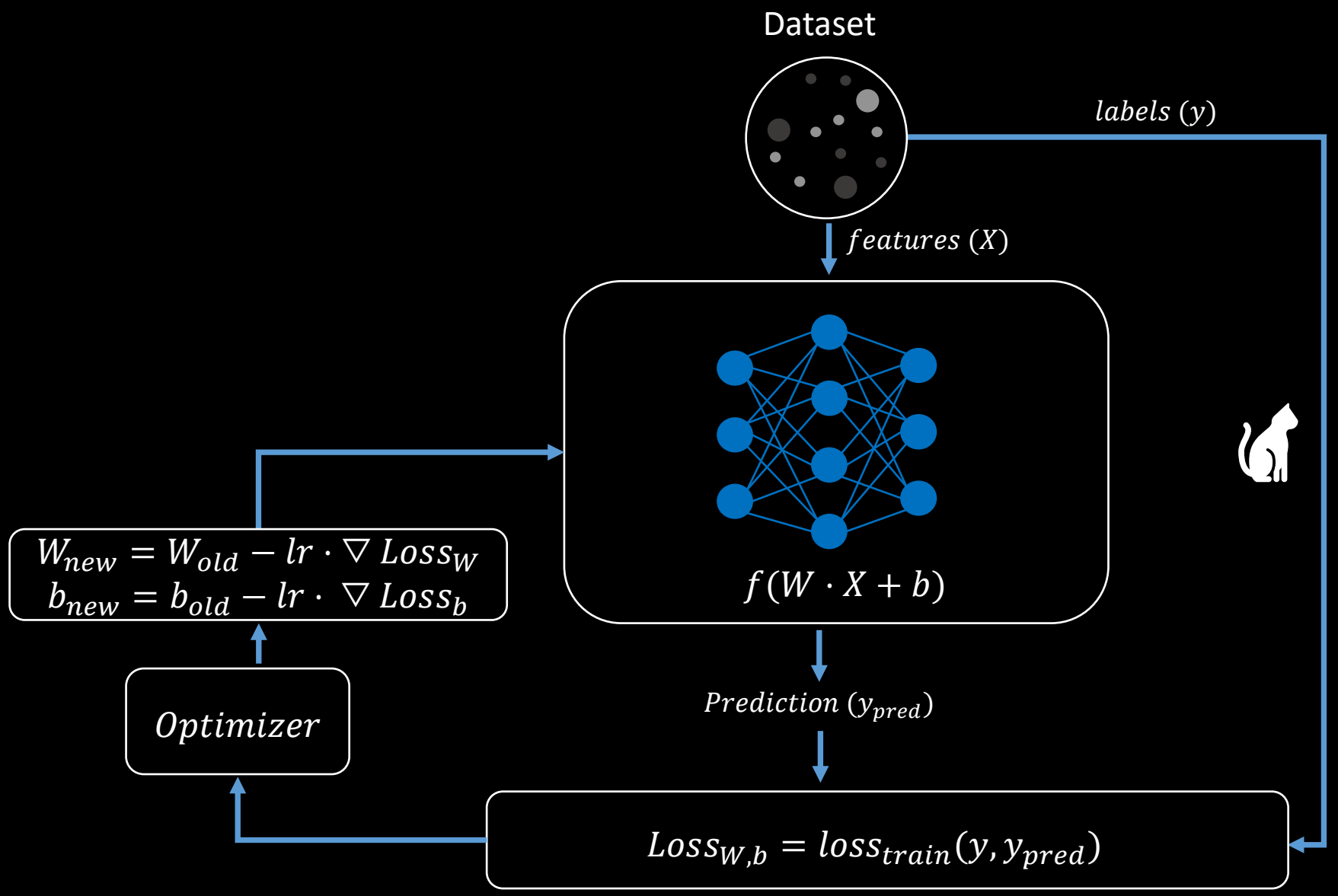
$f()$: *model hypothesis*

W, b: *model parameters*

lr: *learning rate*

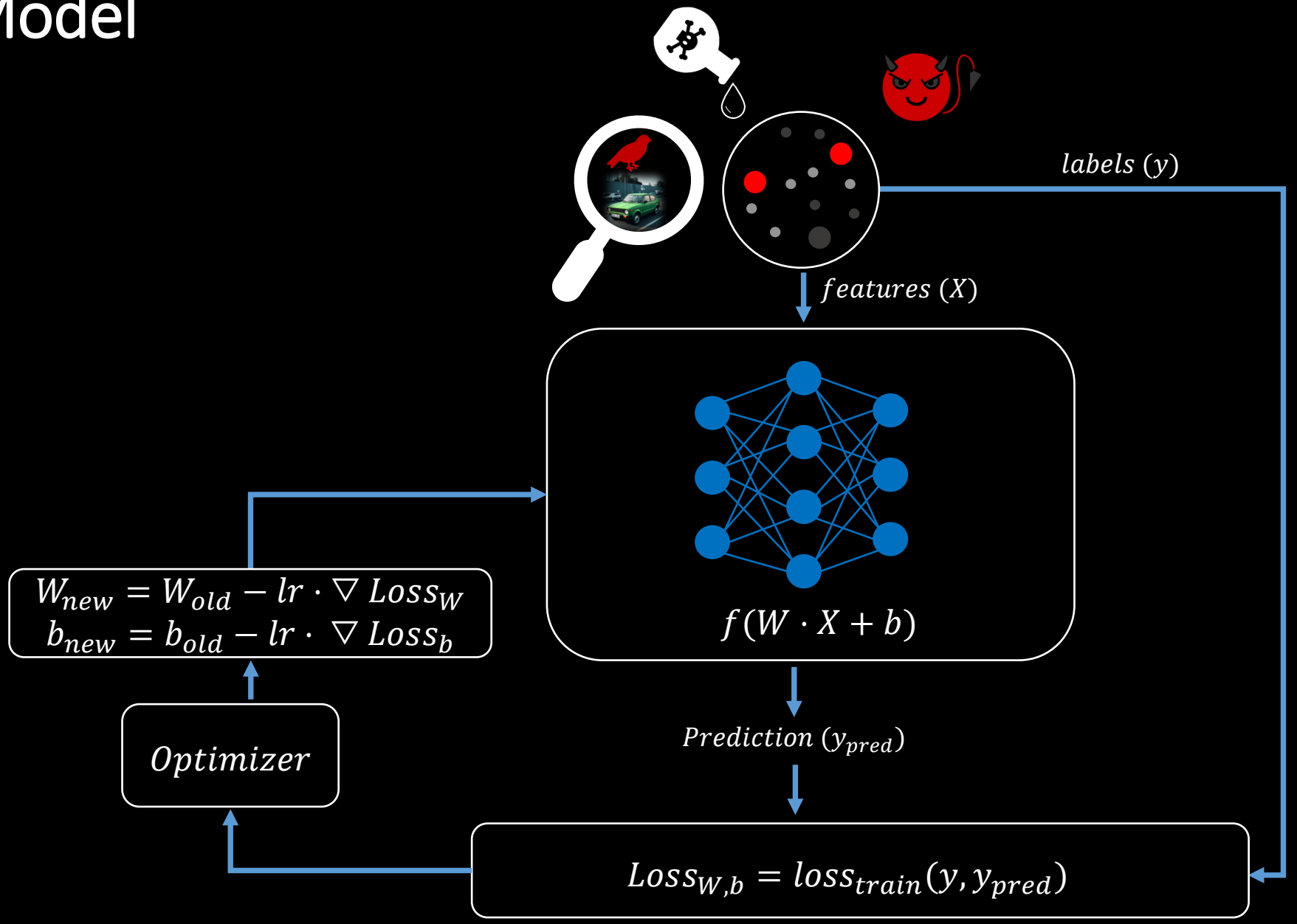
∇: *gradient*

Poisoning Local Model



$f()$: model hypothesis
W, b: model parameters
lr: learning rate
∇: gradient

Poisoning Local Model



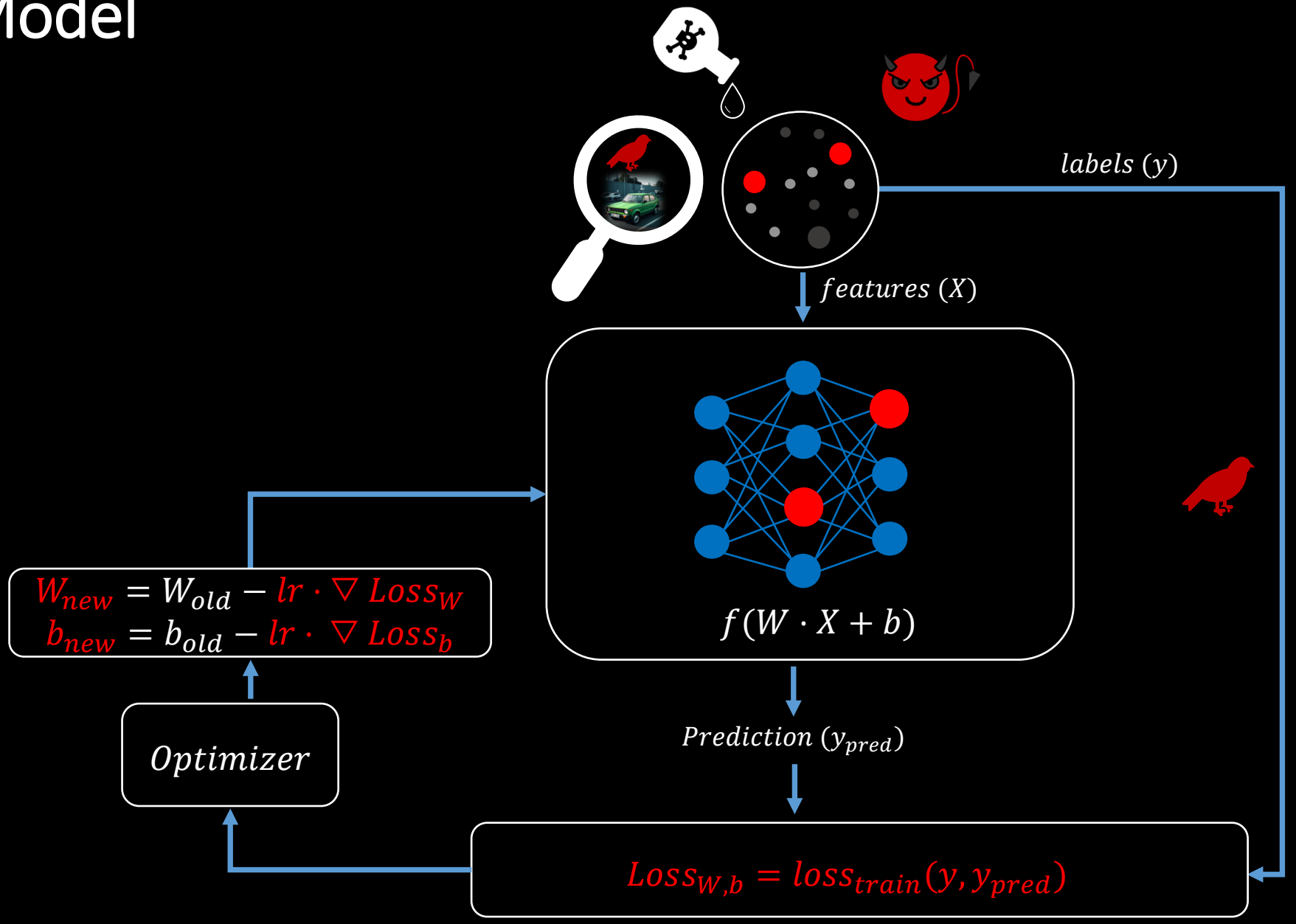
$f()$: *model hypothesis*

W, b: *model parameters*

lr: *learning rate*

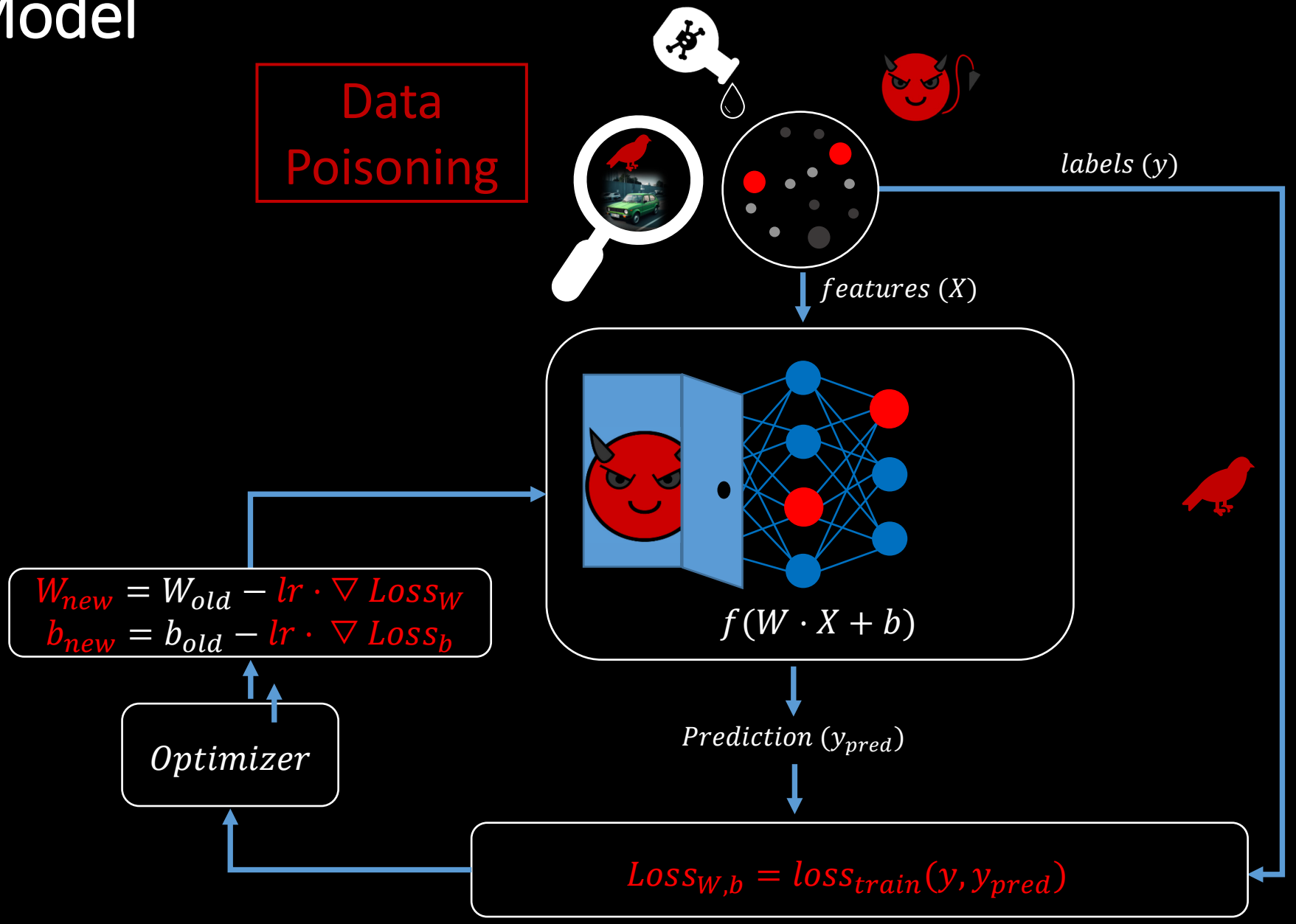
∇: *gradient*

Poisoning Local Model



$f()$: model hypothesis
W, b: model parameters
lr: learning rate
∇: gradient

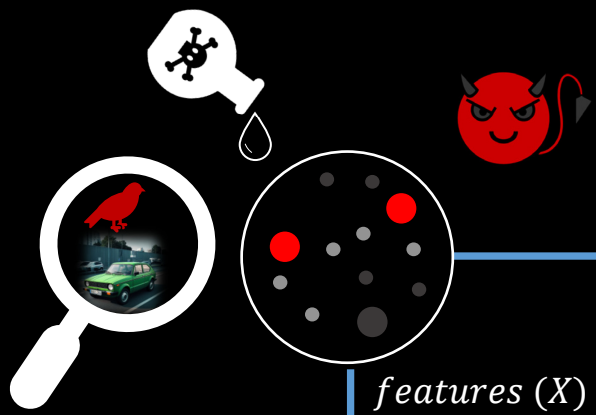
Poisoning Local Model



$f()$: model hypothesis
W, b: model parameters
lr: learning rate
∇: gradient

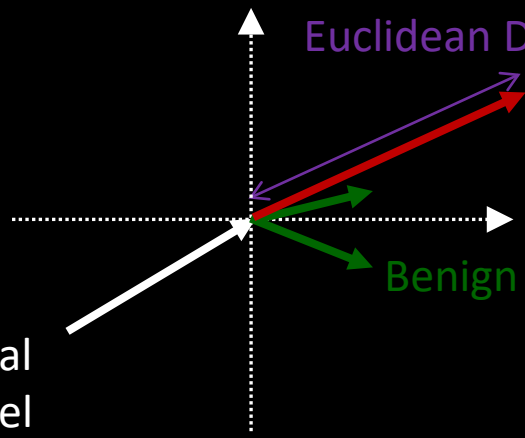
Poisoning Local Model

Data Poisoning

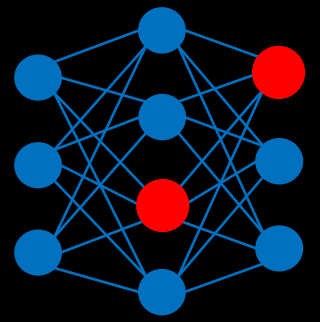


$labels\ (y)$

$features\ (X)$



$$W_{new} = W_{old} - lr \cdot \nabla Loss_W$$
$$b_{new} = b_{old} - lr \cdot \nabla Loss_b$$



$f(W \cdot X + b)$

$Optimizer$

$Prediction\ (y_{pred})$

$Loss_{W,b} = loss_{train}(y, y_{pred})$

$f()$: *model hypothesis*

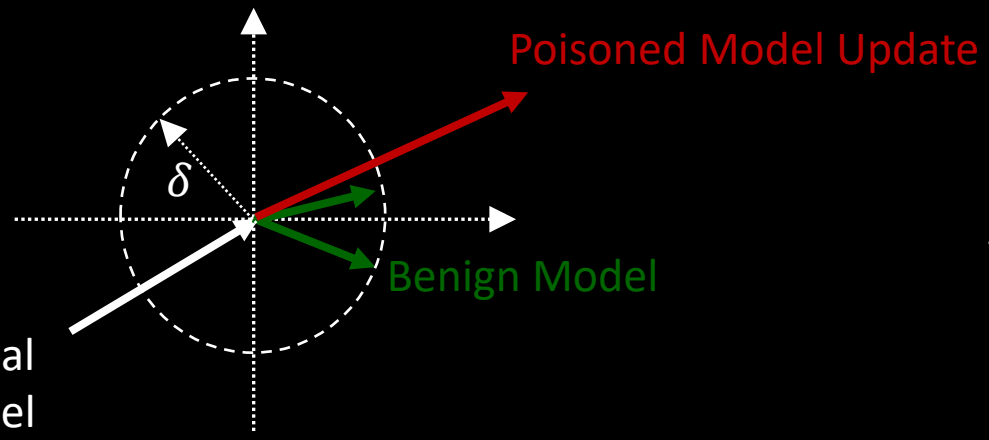
W, b: *model parameters*

lr: *learning rate*

∇: *gradient*

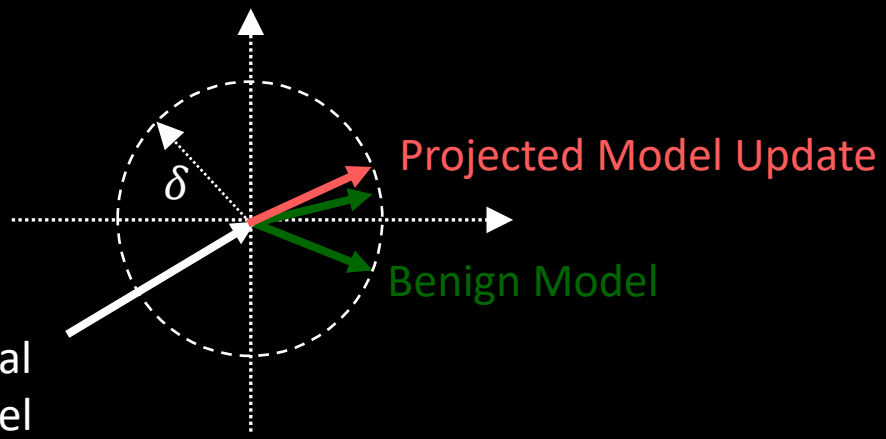
Projected Gradient Descent

Model ~~Data~~ Poisoning



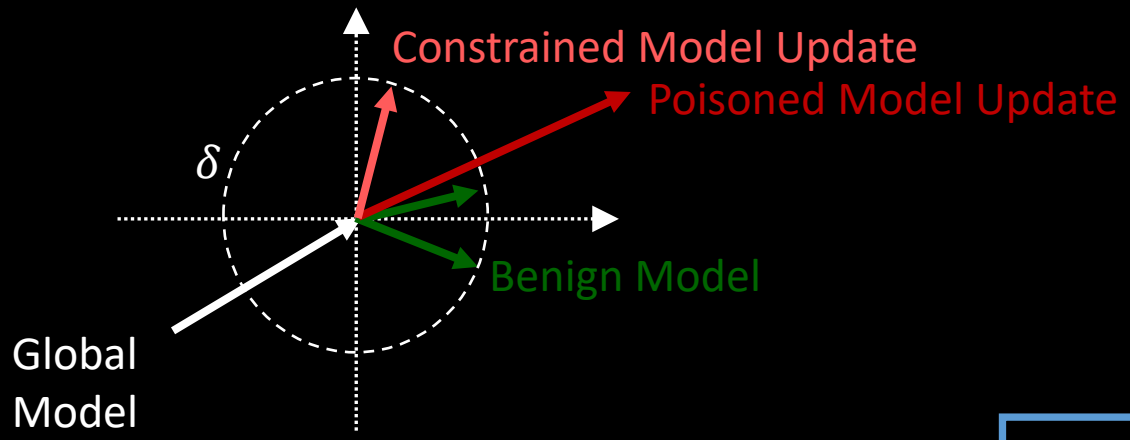
Attack Budget: $\delta = \| \text{Benign Model} - \text{Global Model} \|$

Downscale Poisoned Model

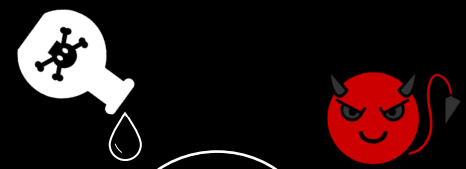


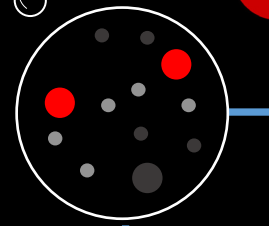
Projection: $\delta \geq \| \text{Poisoned Model} - \text{Global Model} \|$

Loss Constrain



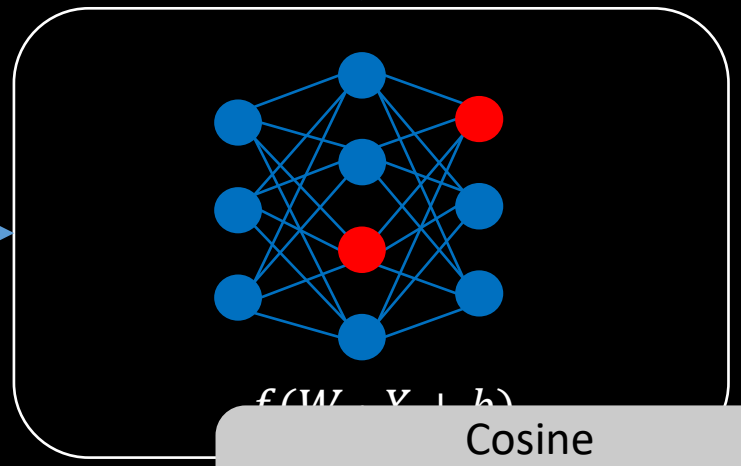
Model Data Poisoning





$labels\ (y)$

$features\ (X)$



$f(W \cdot X + b)$

$W_{new} = W_{old} - lr \cdot \nabla Loss_W$

$b_{new} = b_{old} - lr \cdot \nabla Loss_b$

$Optimizer$

Cosine
Euclidean Distance
Any Metric used by Defense

$Loss_{W,b} = loss_{train}(y, y_{pred}) + loss_{Evasion}$

$f()$: model hypothesis

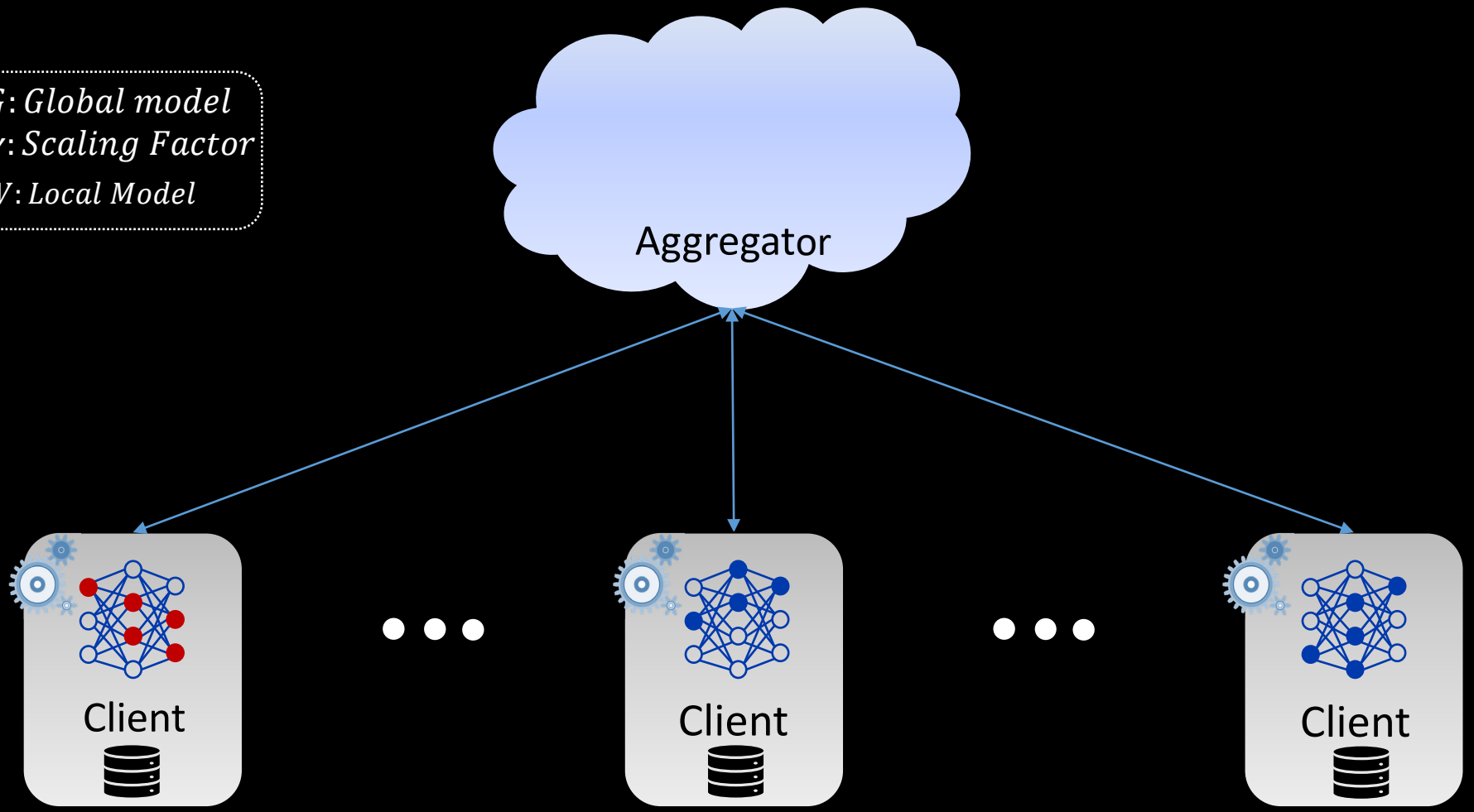
W, b: model parameters

lr: learning rate

∇: gradient

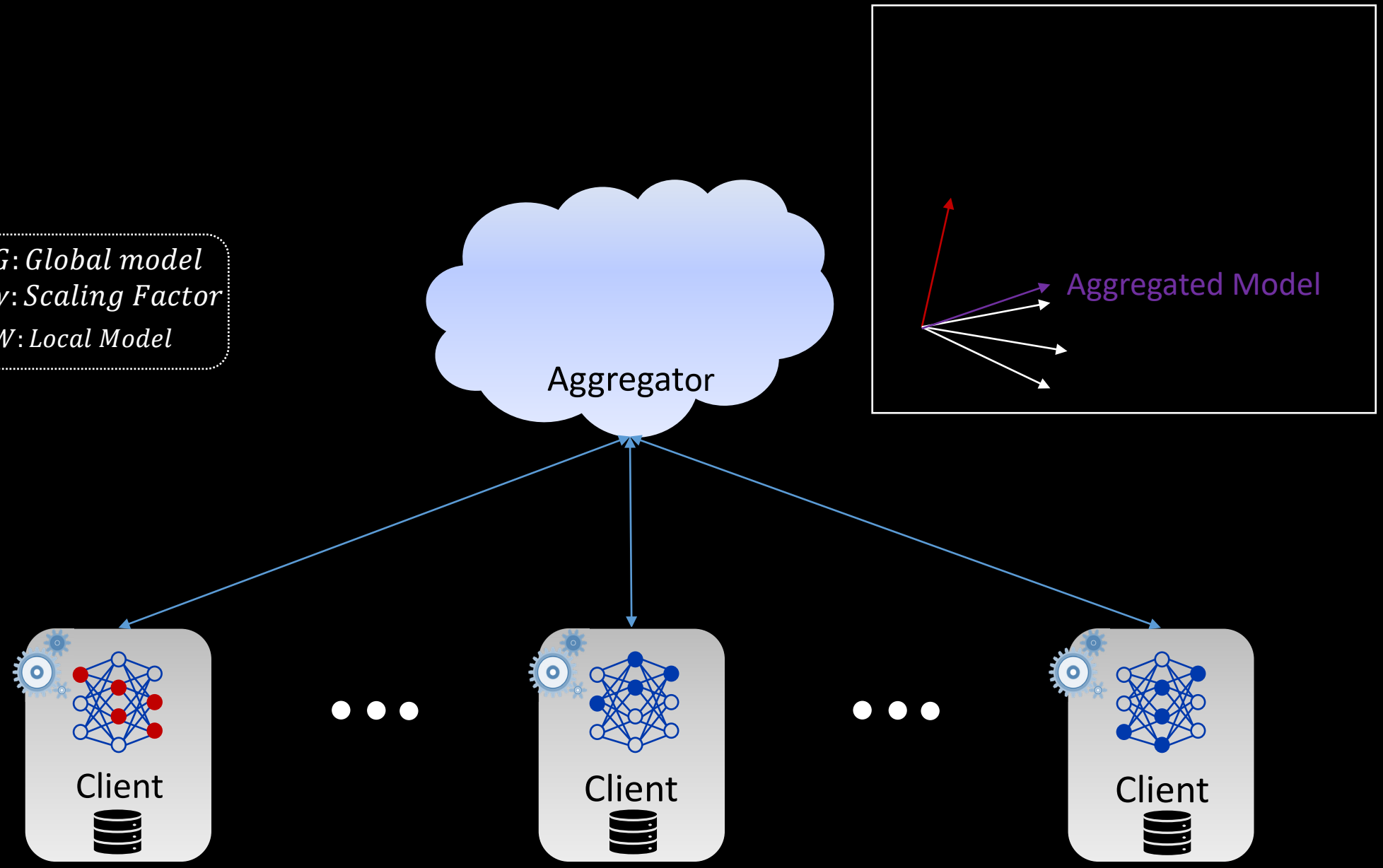
Scaling

G: *Global model*
γ: *Scaling Factor*
W: *Local Model*



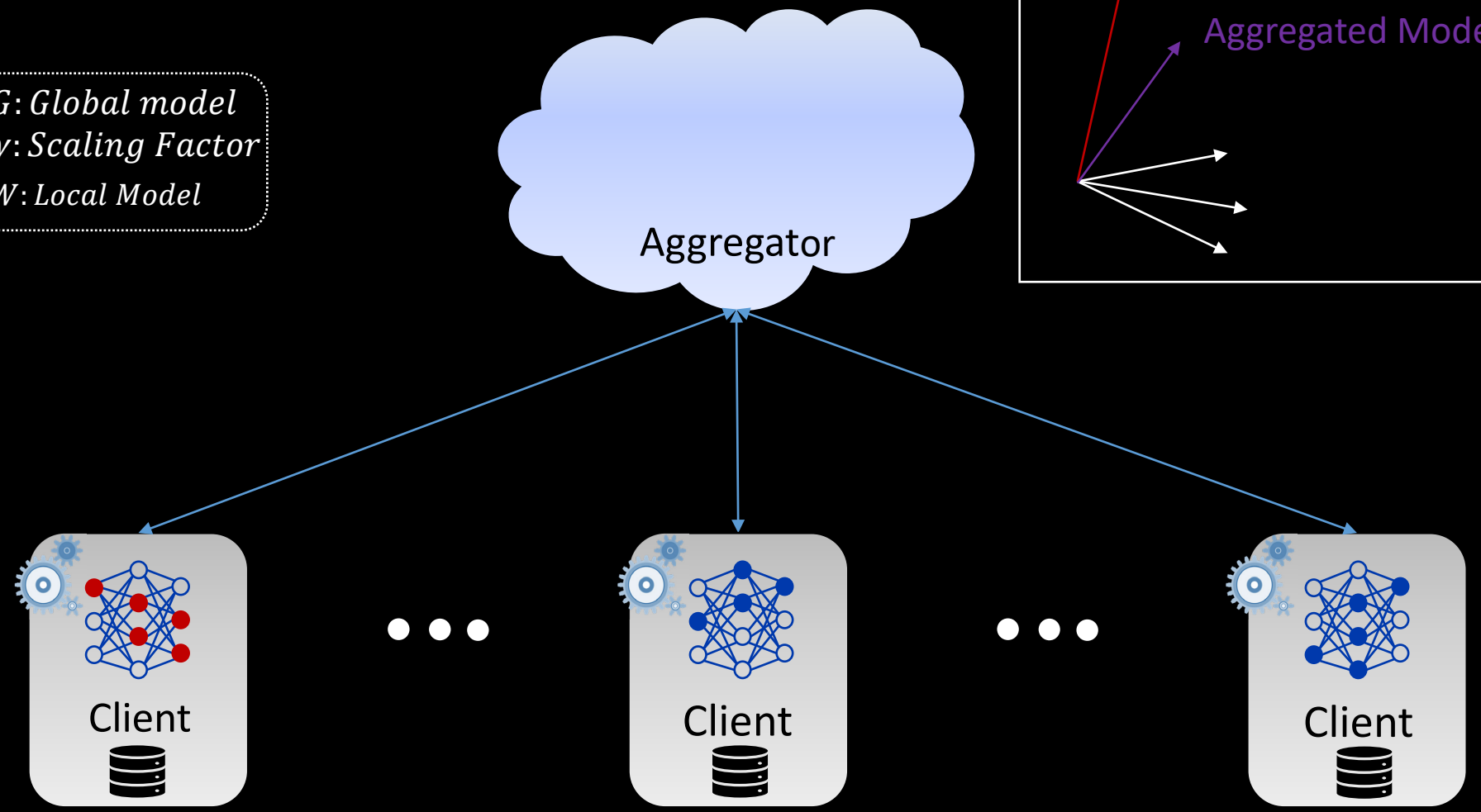
Scaling

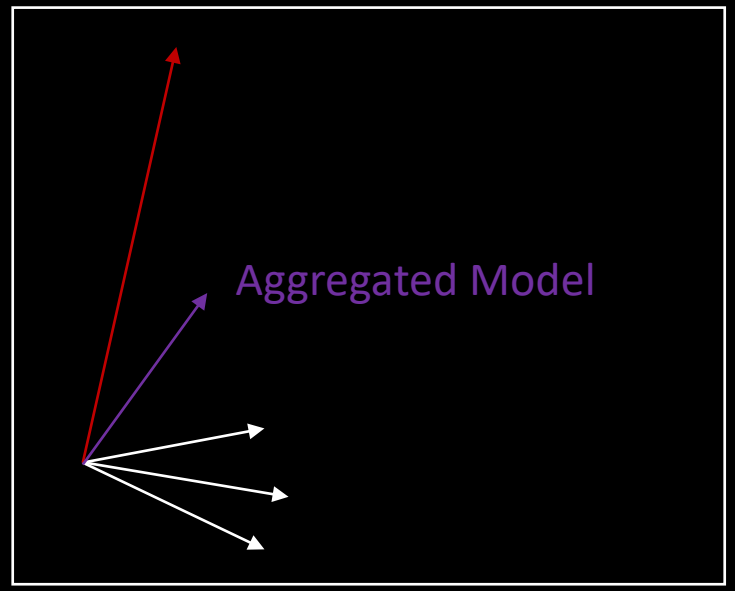
G: *Global model*
γ: *Scaling Factor*
W: *Local Model*



Scaling

G: *Global model*
γ: *Scaling Factor*
W: *Local Model*



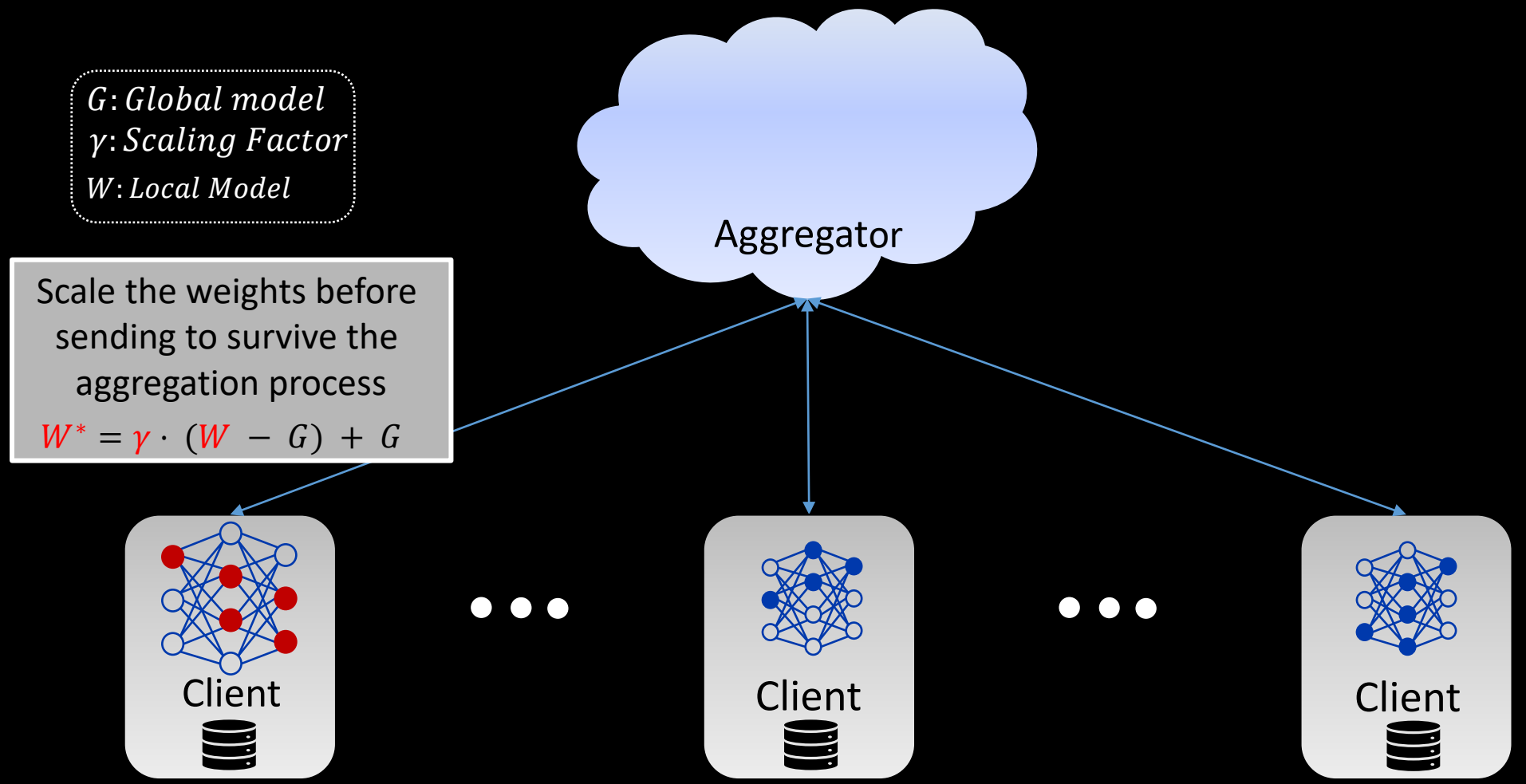


Scaling

G: *Global model*
γ: *Scaling Factor*
W: *Local Model*

Scale the weights before sending to survive the aggregation process

$$W^* = \gamma \cdot (W - G) + G$$

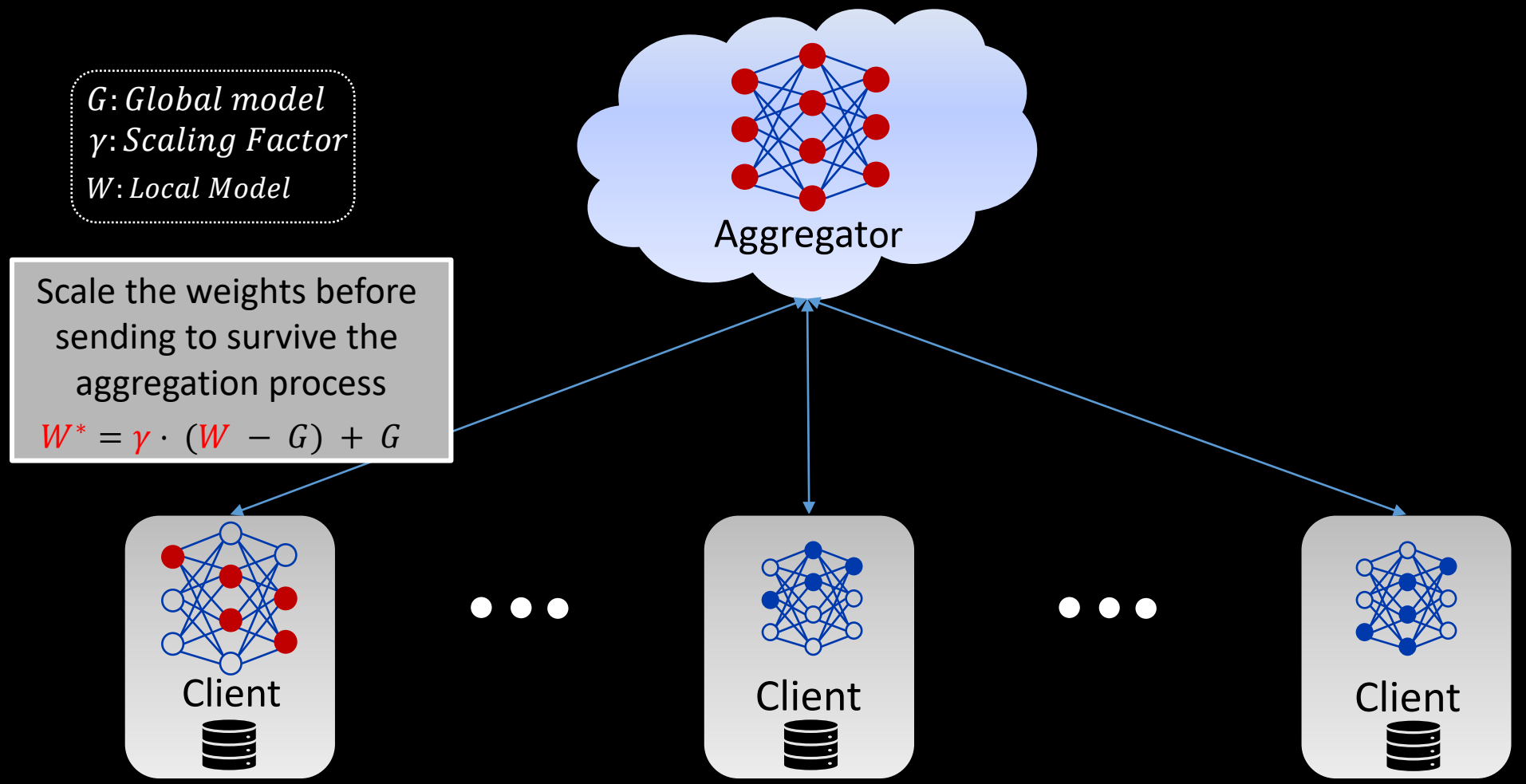


Scaling

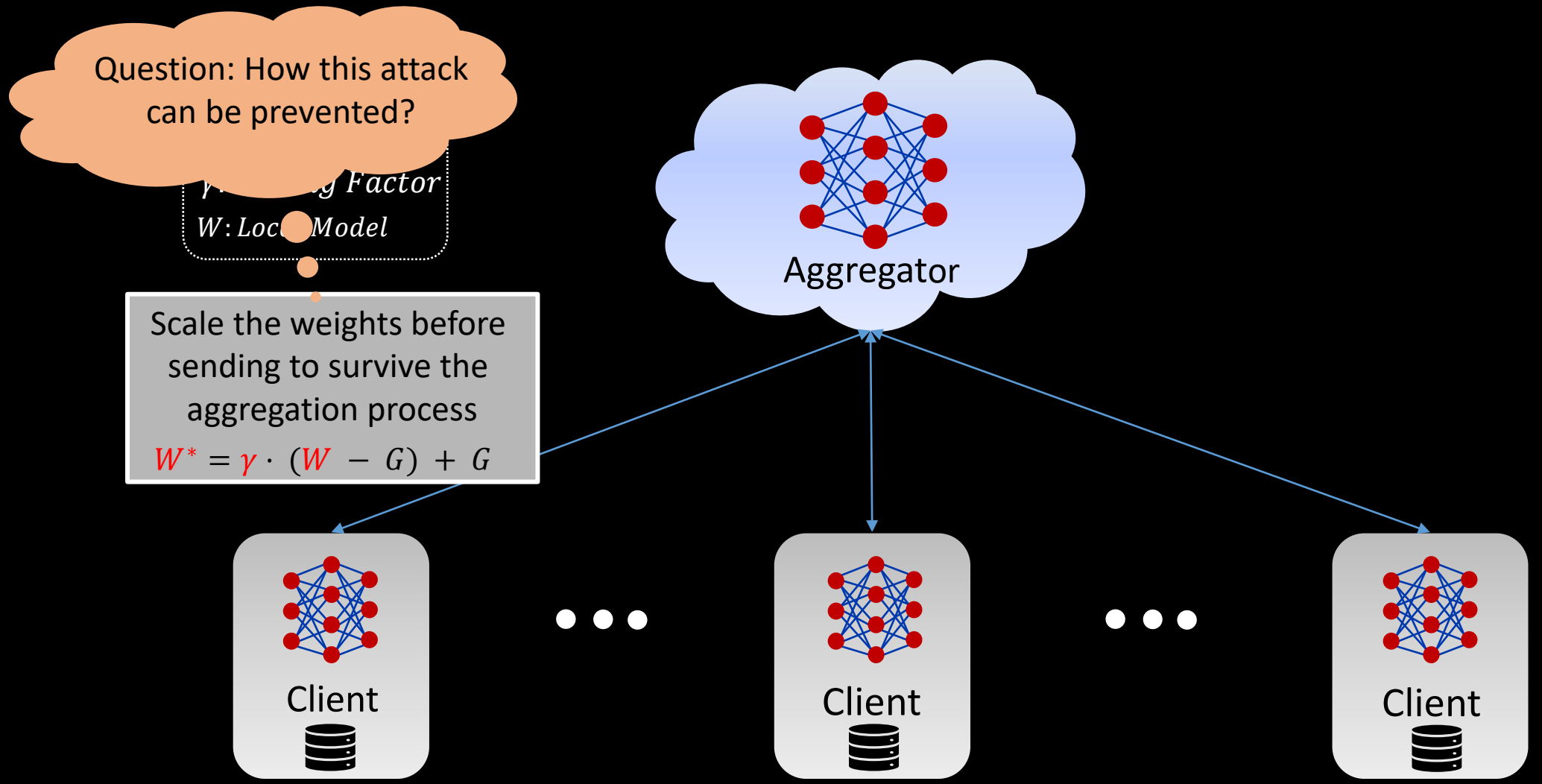
G: *Global model*
γ: *Scaling Factor*
W: *Local Model*

Scale the weights before sending to survive the aggregation process

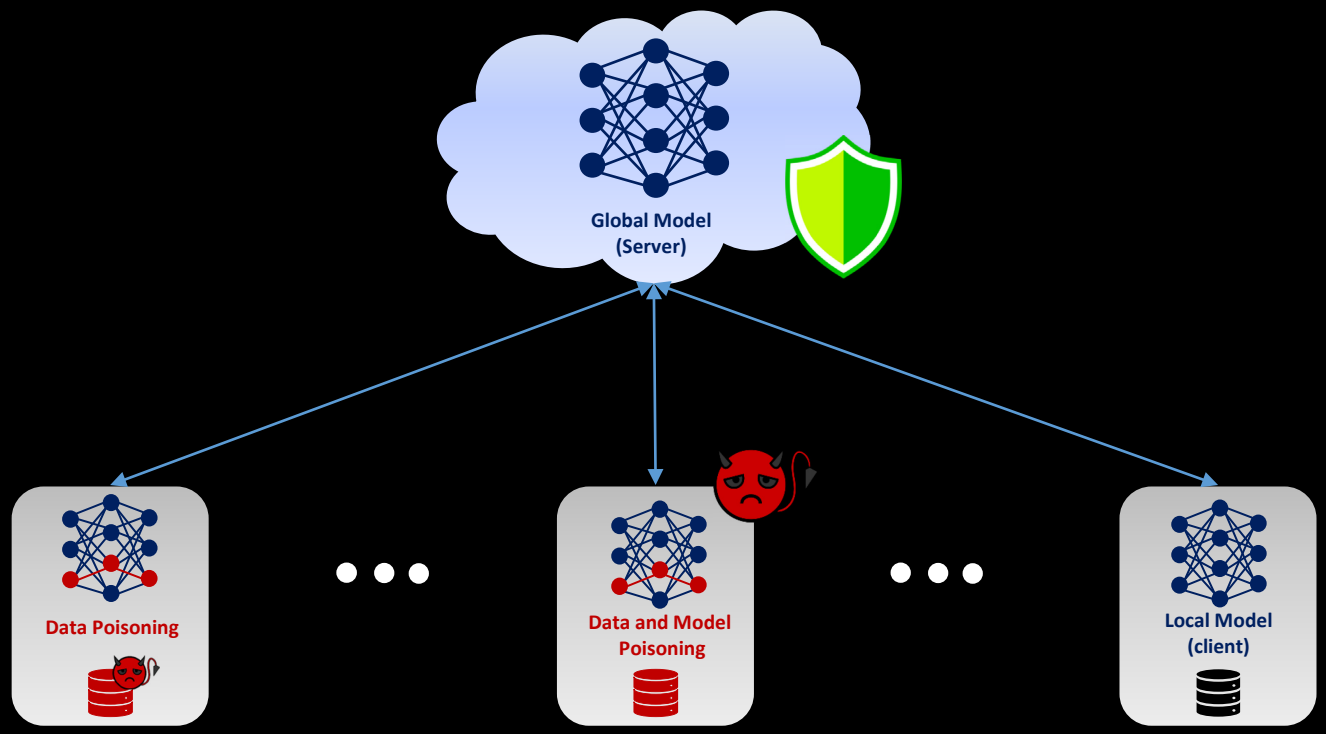
$$W^* = \gamma \cdot (W - G) + G$$



Scaling

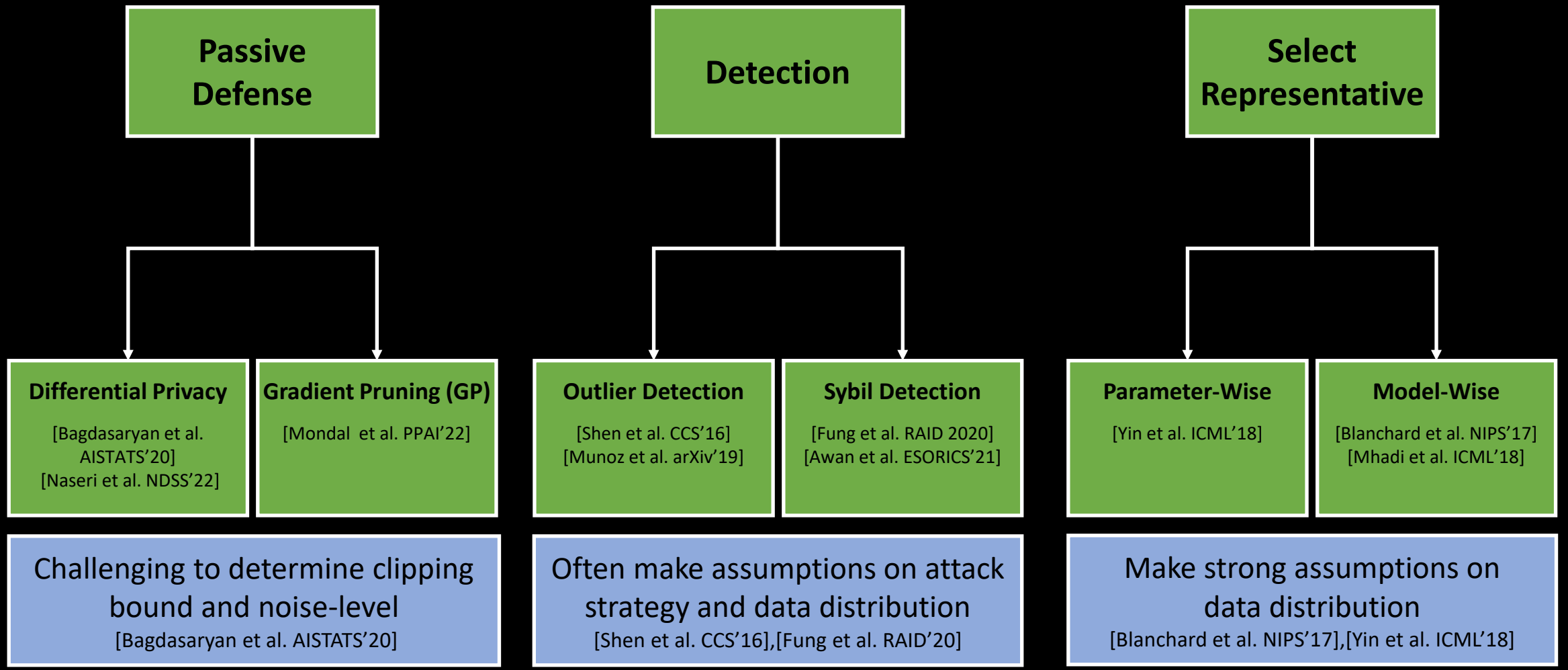


Building Poisoning Resilient FL Systems

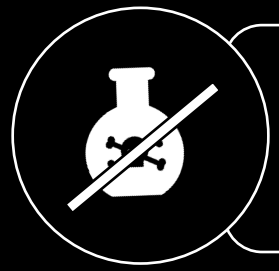


No Pressure!

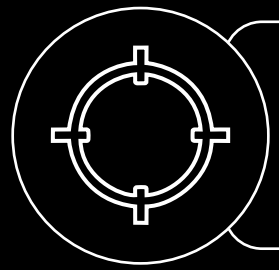
Existing Defenses Against Backdoor Attacks



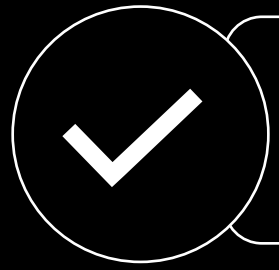
Advantages of Detection Approaches



- Aggregated model is backdoor free, if all poisoned models are detected

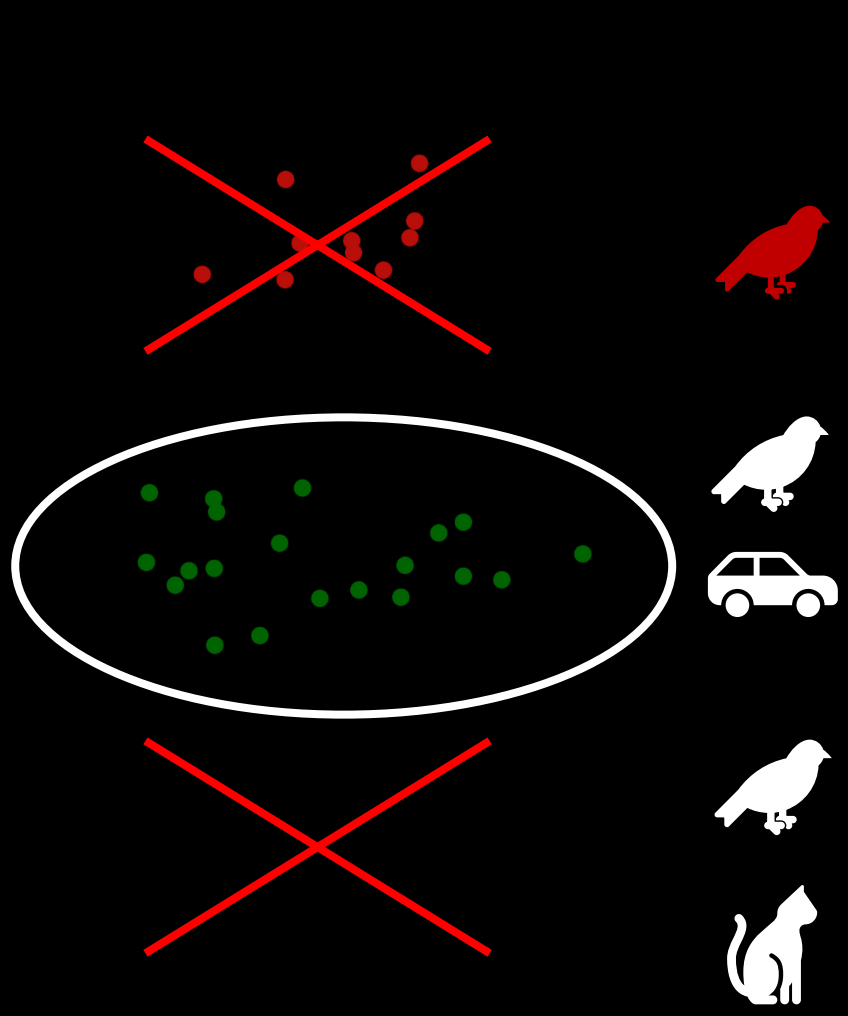


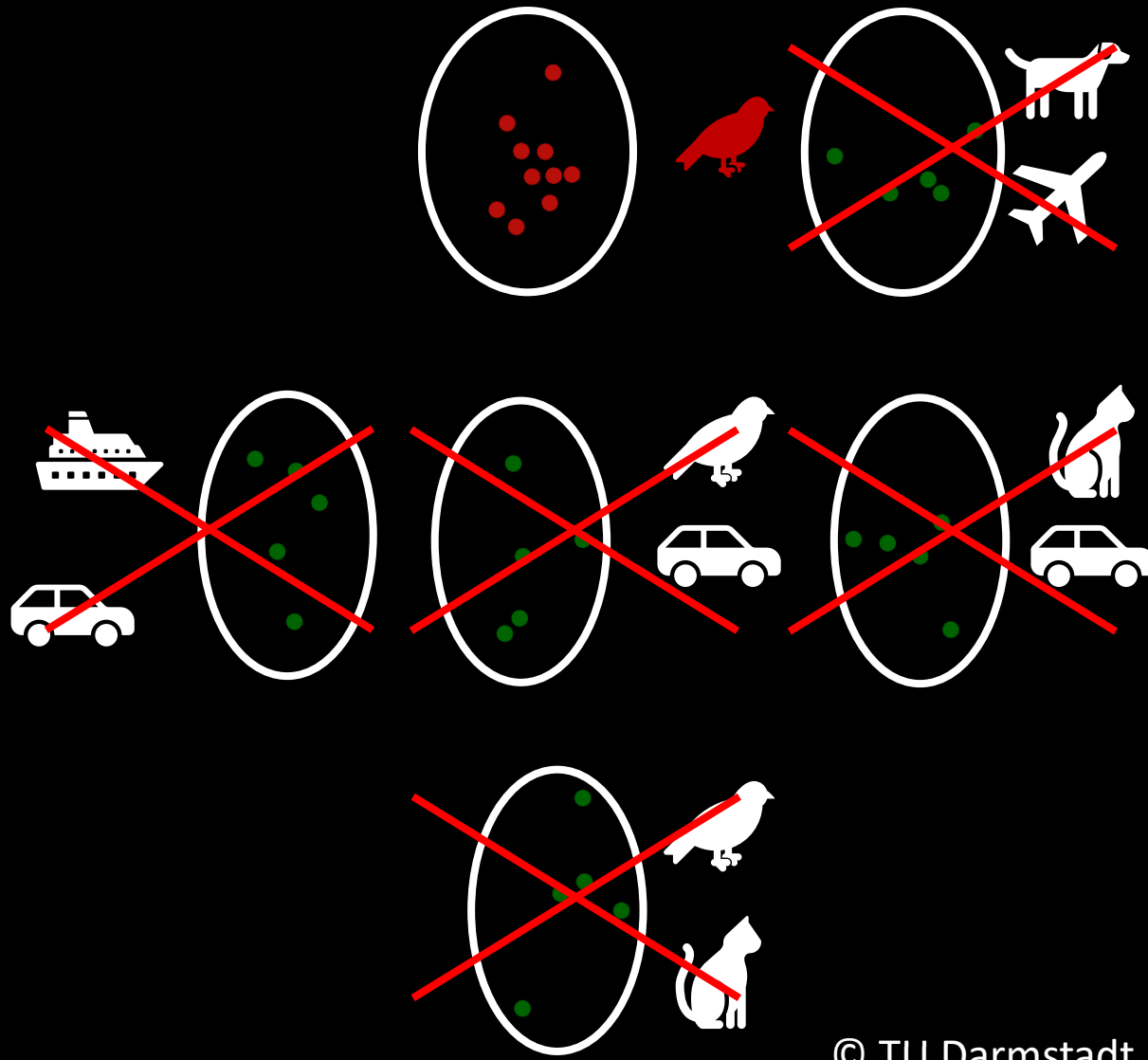
- Attackers can be identified
- Allows for permanently banning attackers



- Utility of model not reduced, if no benign model is excluded

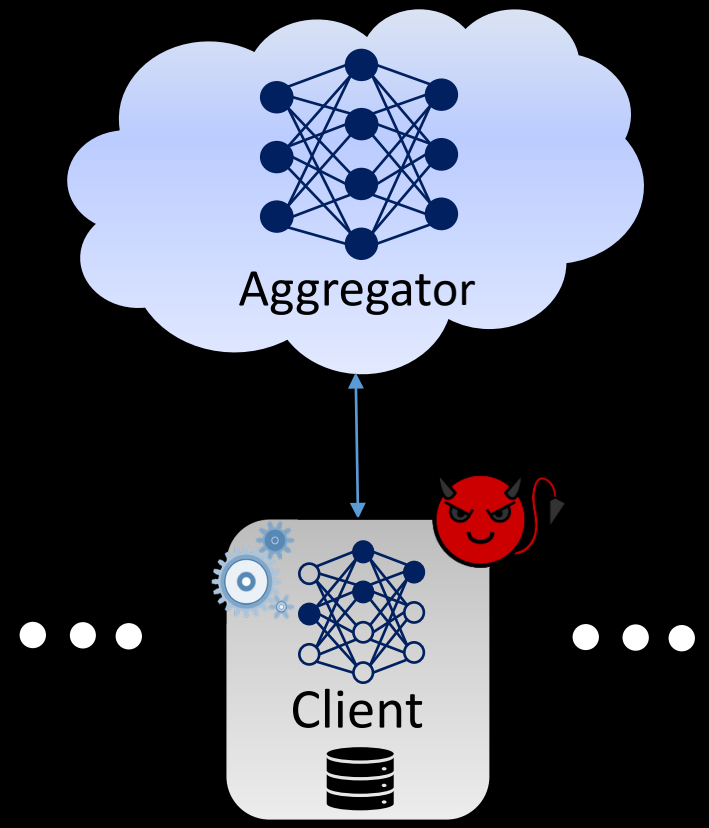
Detecting Poisoned Models in non-IID Scenarios





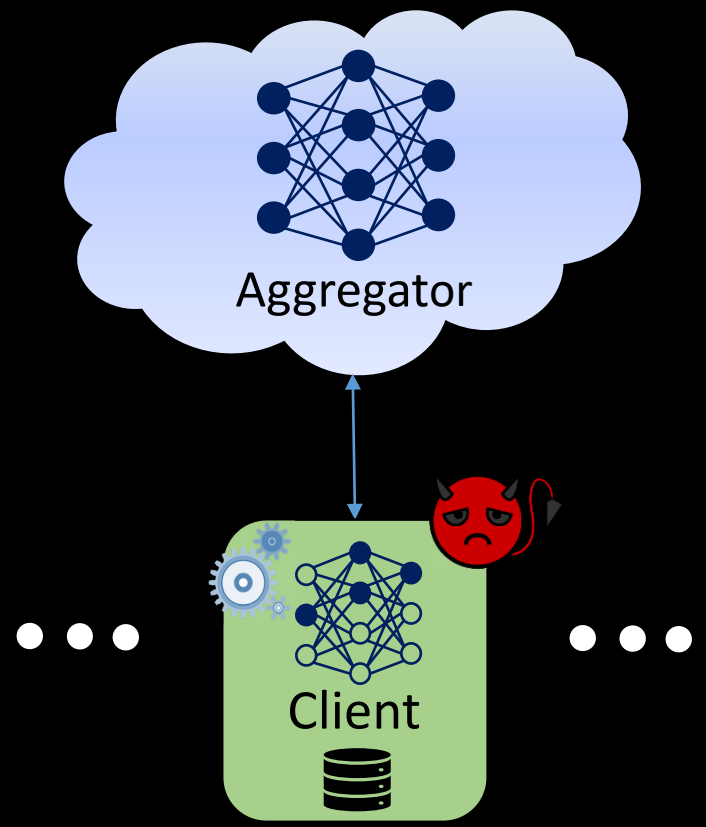
Helpfulness of Trusted Execution Environments (TEEs)

- TEEs guarantee correct code execution



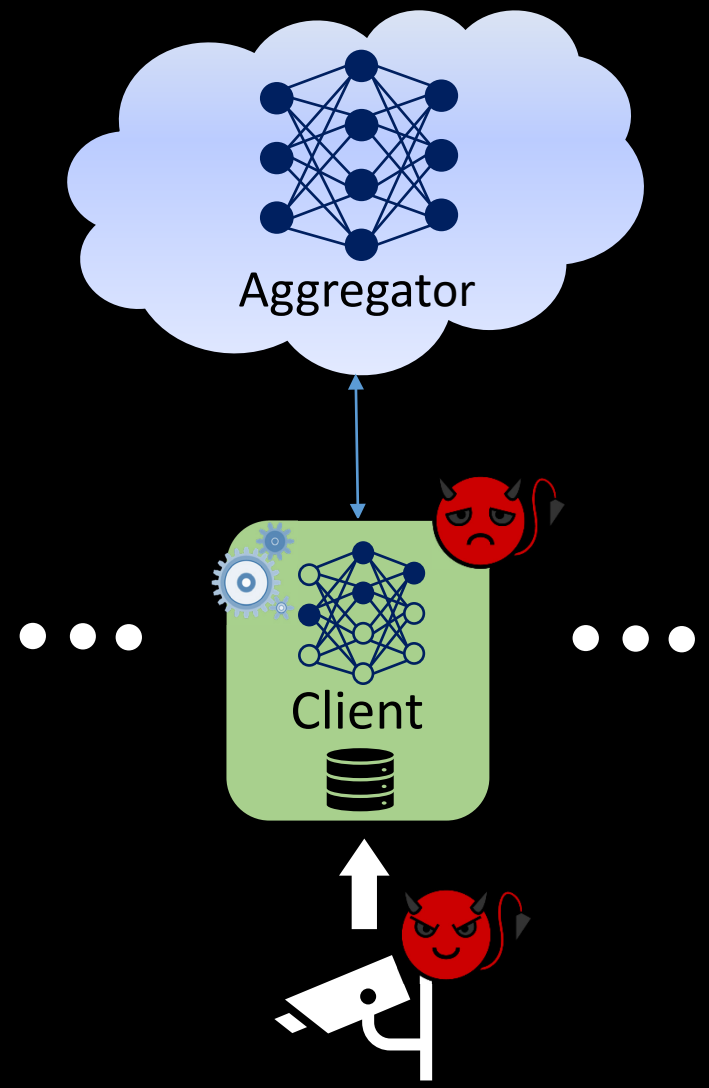
Helpfulness of Trusted Execution Environments (TEEs)

- TEEs guarantee correct code execution
- Effectively prevent clients from intentionally injecting backdoor
- Problem:
  - Slows down training



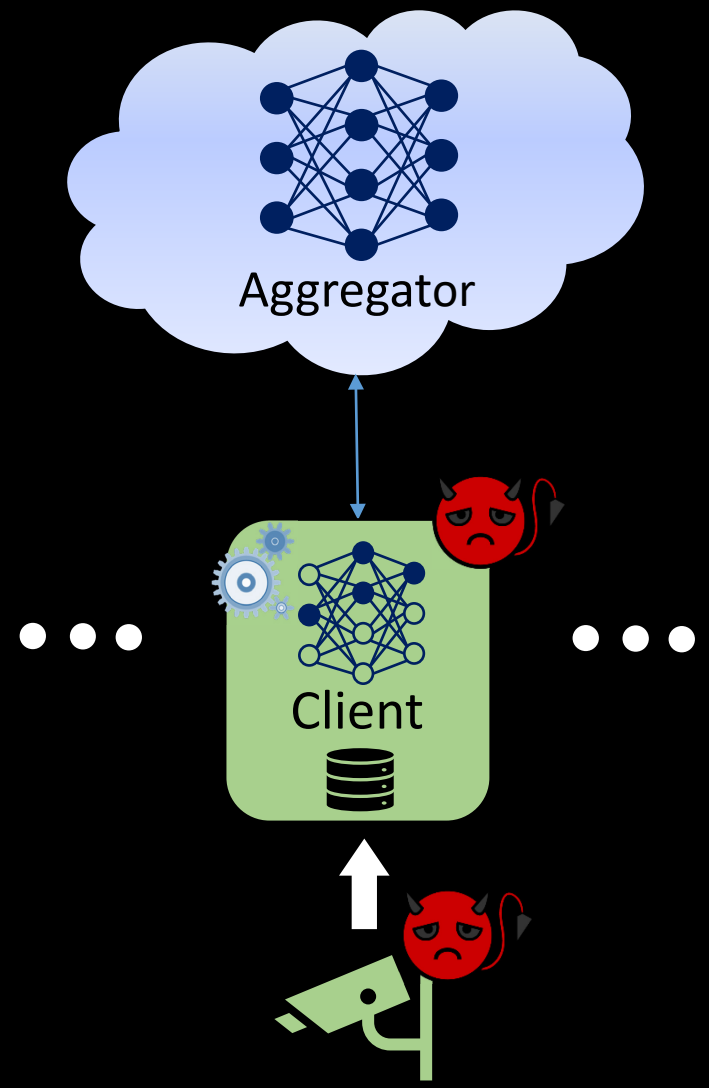
Helpfulness of Trusted Execution Environments (TEEs)

- TEEs guarantee correct code execution
- Effectively prevent clients from intentionally injecting backdoor
- Problem:
  - Slows down training



Helpfulness of Trusted Execution Environments (TEEs)

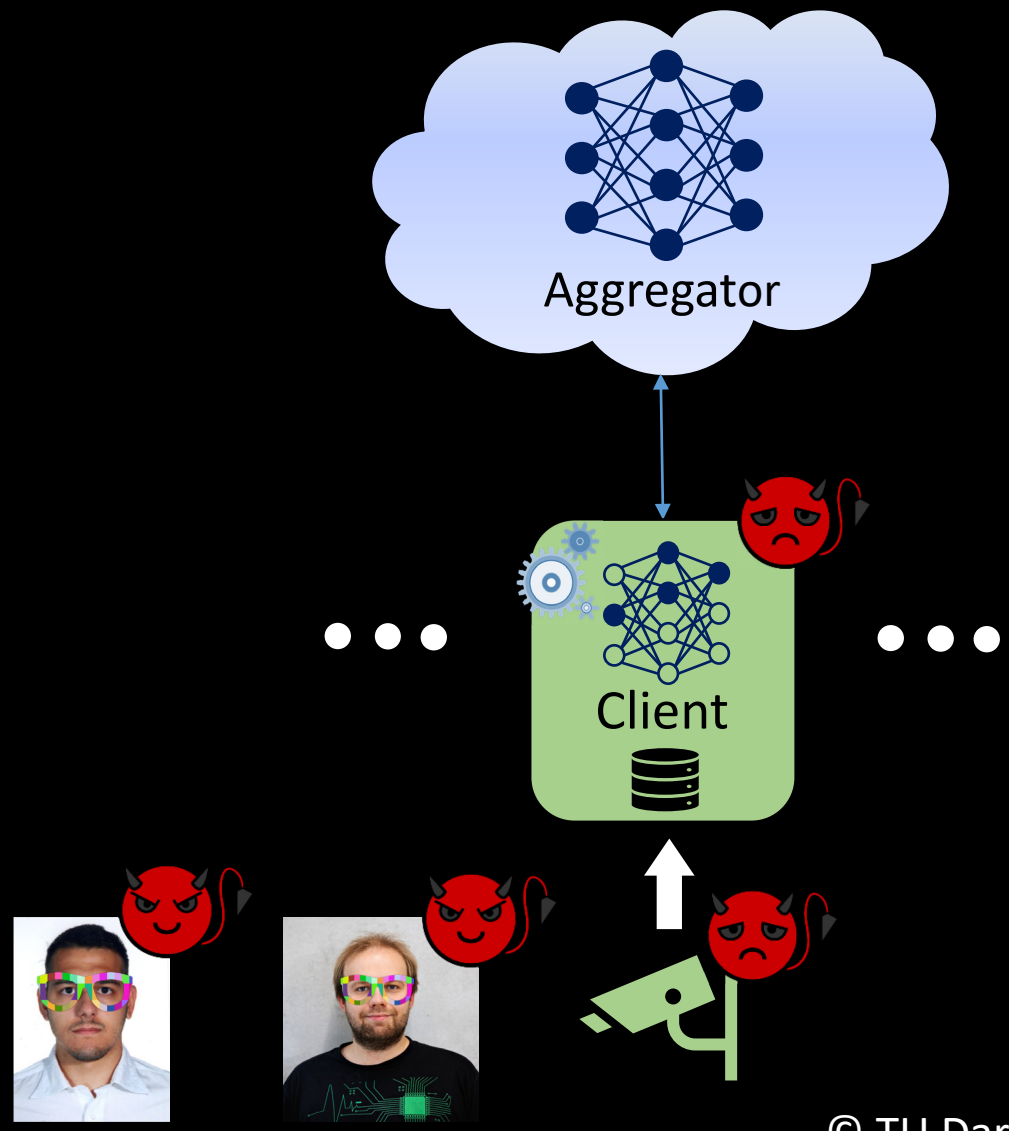
- TEEs guarantee correct code execution
- Effectively prevent clients from intentionally injecting backdoor
- Problem:
  - Slows down training



Helpfulness of Trusted Execution Environments (TEEs)

- TEEs guarantee correct code execution
- Effectively prevent clients from intentionally injecting backdoor
- Problem:
  - Slows down training
  - Input to TEE not controllable

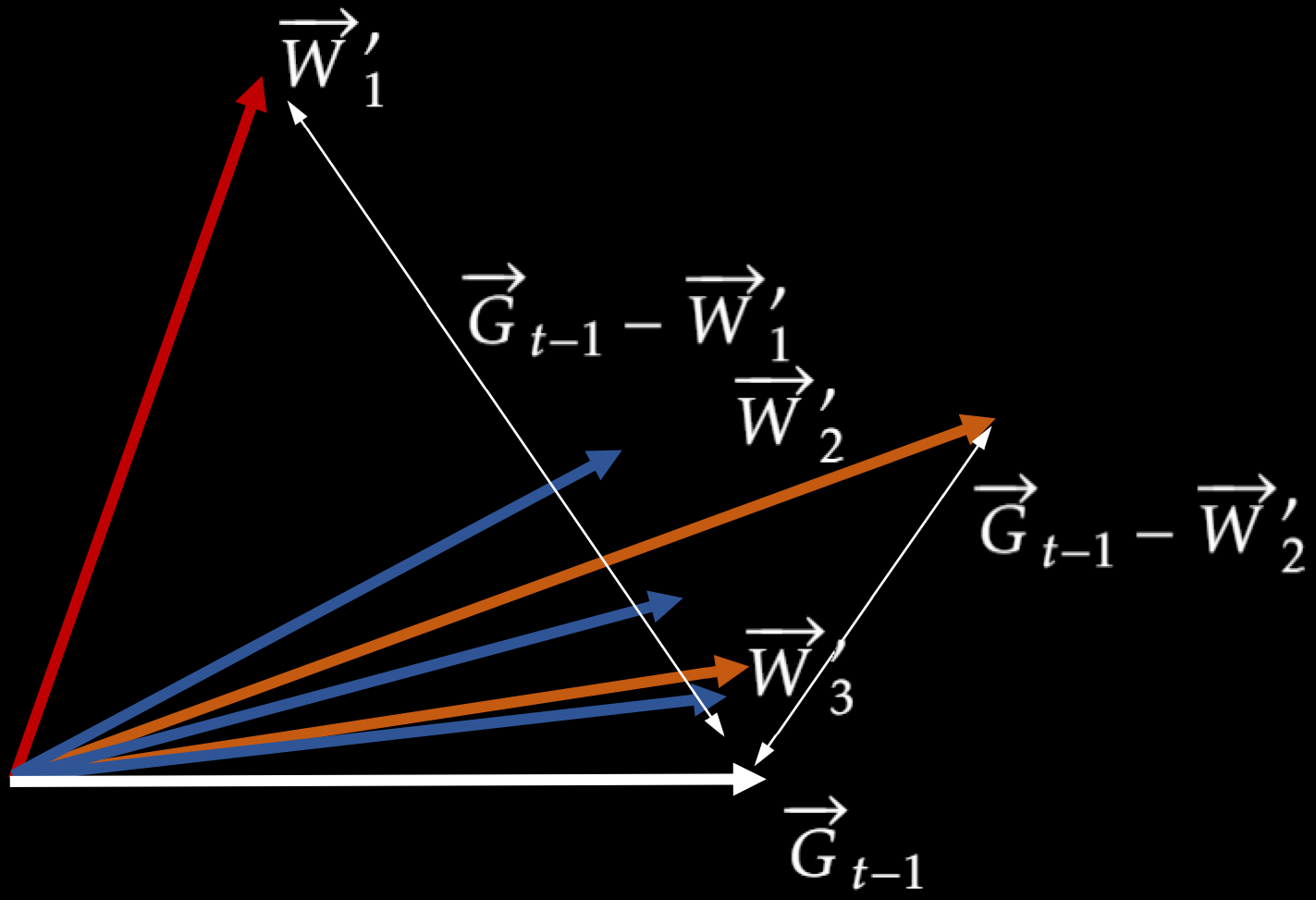
→ TEEs should only be used for privacy protection



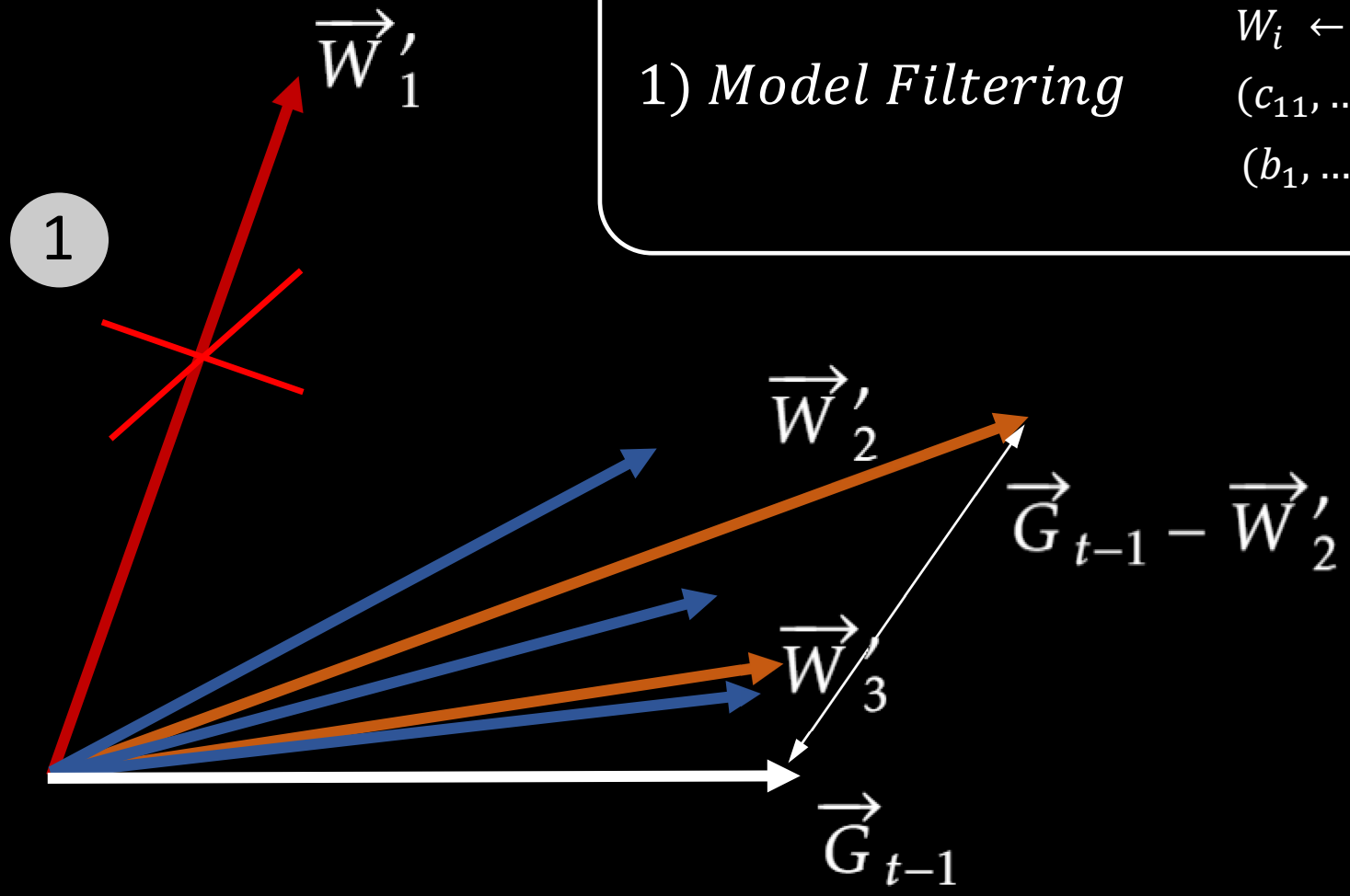
FLAME: Taming Backdoors in Federated Learning

Thien-Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samual Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, Thomas Schneider. **USENIX Security 2022**

High-Level Idea



High-Level Idea



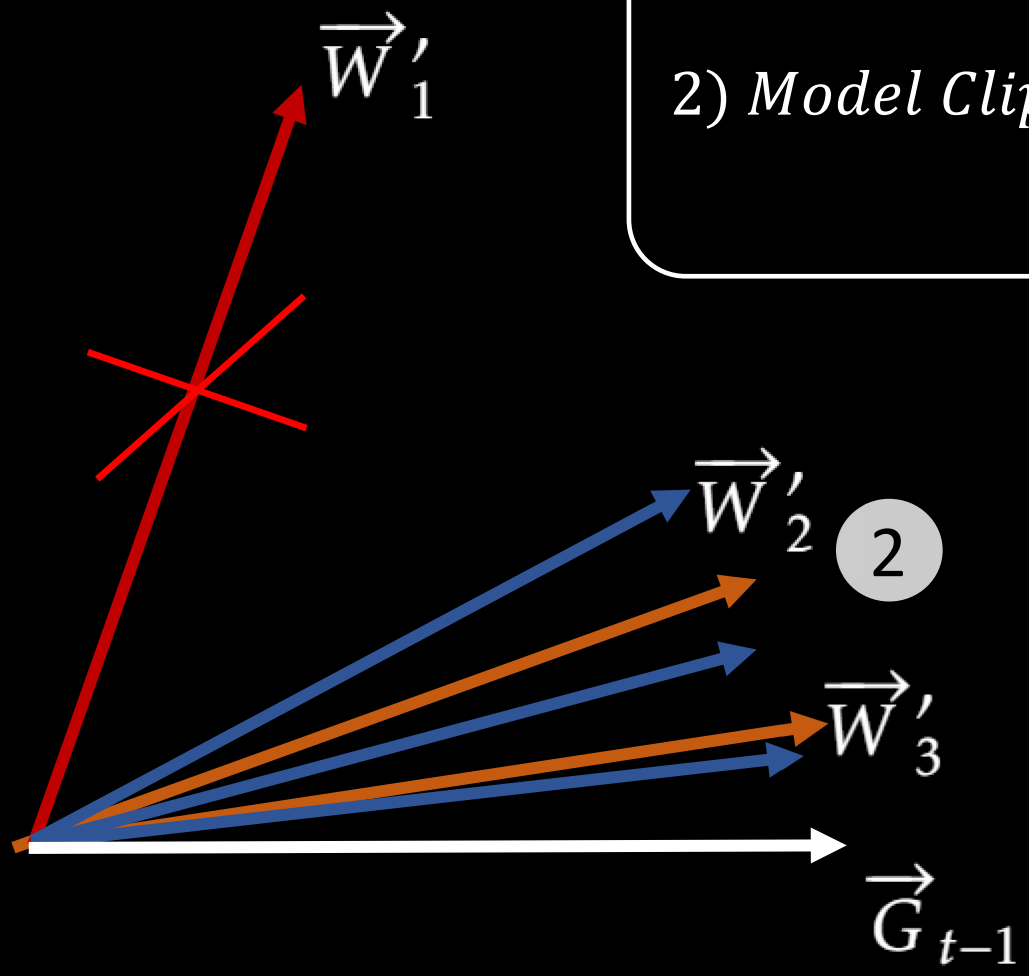
1) *Model Filtering*

$W_i \leftarrow Client\_Update(G_{t-1})$

$(c_{11}, \ldots, c_{nn}) \leftarrow Cosine\_Distance(W_1, \ldots, W_n)$

$(b_1, \ldots, b_L) \leftarrow Clustering(c_{11}, \ldots, c_{nn})$

High-Level Idea



2) *Model Clipping*

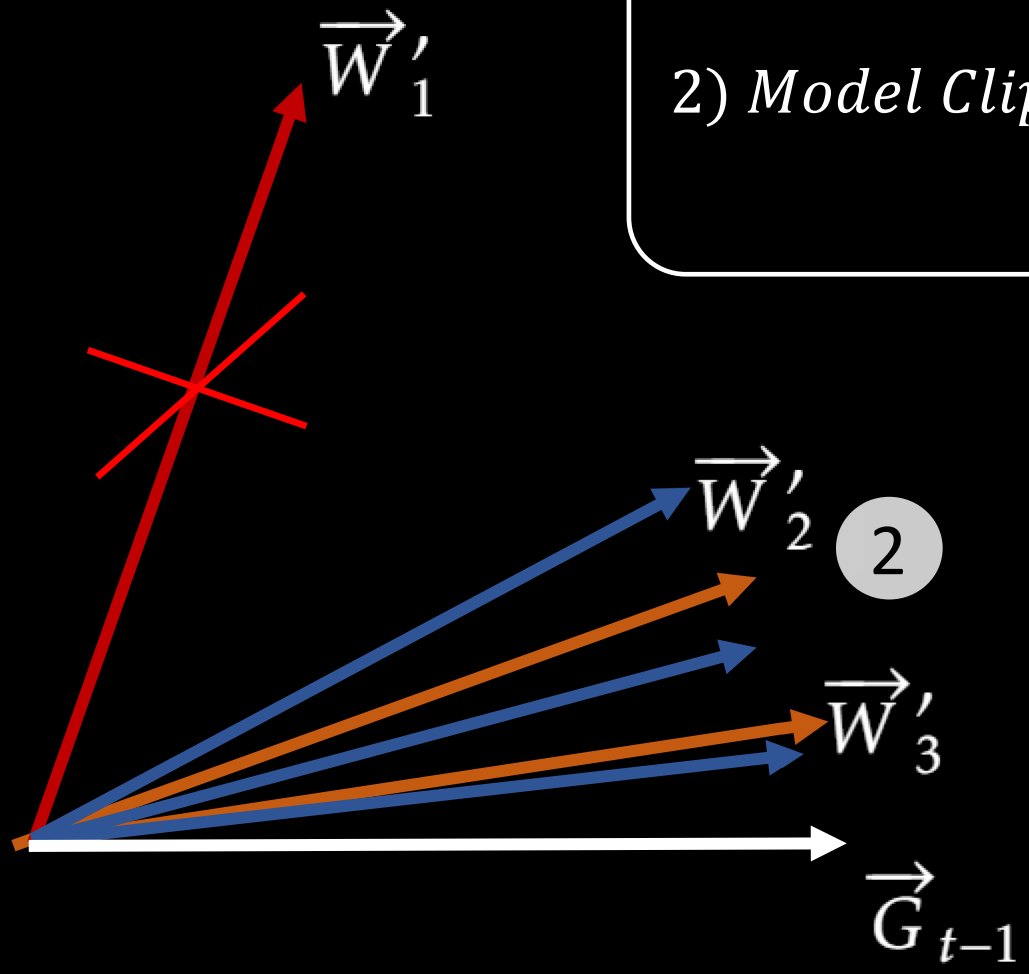
$$(e_1, \dots, e_n) \leftarrow Euclidean\_Distance(G_{t-1}, (W_1, \dots, W_n))$$

$$S_t \leftarrow Median(e_1, \dots, e_n)$$

$$W_j \leftarrow G_{t-1} + (W_j - G_{t-1}) * Min\left(1, \frac{S_t}{e_j}\right) \forall \in \{b_1, \dots, b_L\}$$

$$\overrightarrow{G}_{t-1} - \overrightarrow{W}'_2$$

High-Level Idea



$\overrightarrow{G}_{t-1} - \overrightarrow{W}'_2$

2) *Model Clipping*

$$(e_1, \dots, e_n) \leftarrow Euclidean\_Distance(G_{t-1}, (W_1, \dots, W_n))$$

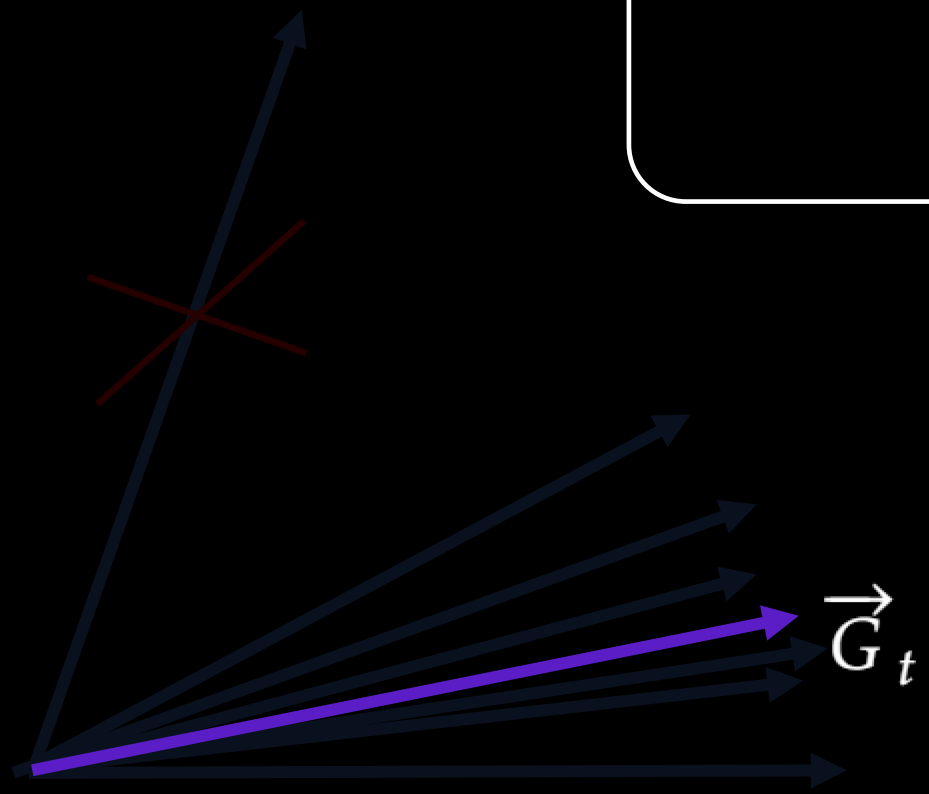
$$S_t \leftarrow Median(e_1, \dots, e_n)$$

$$W_j \leftarrow G_{t-1} + (W_j - G_{t-1}) * Min\left(1, \frac{S_t}{e_j}\right) \forall \in \{b_1, \dots, b_L\}$$

Question: Why using all models' L2-norms, including removed updates?

High-Level Idea

$$G_t \leftarrow \sum_{j \in \{b_1, \ldots, b_L\}} \frac{W_j}{L}$$

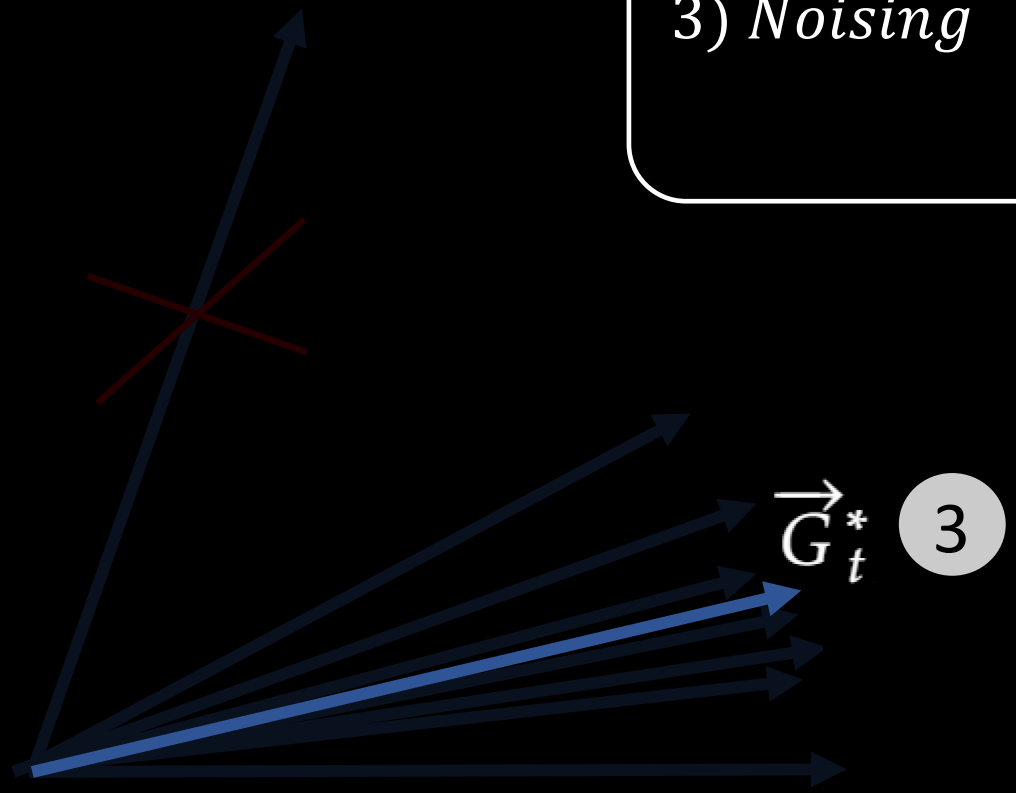


High-Level Idea

3) *Noising*

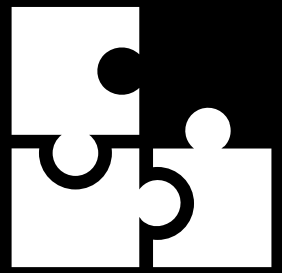
$$G_t \leftarrow \sum_{j \in \{b_1,\ldots,b_L\}} \frac{W_j}{L}$$

$$G_t^* \leftarrow G_t + N(O, \sigma_t^2), \sigma_t \leftarrow \frac{S_t \cdot \sqrt{2 \ln(\frac{1.25}{\delta})}}{\varepsilon}$$



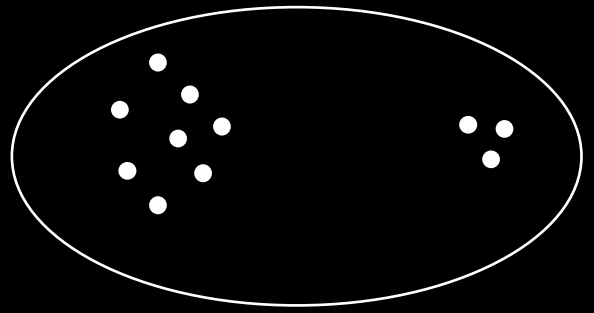
Frequent Flaws when Attacking FLAME

Skip Defense Layers



Implement only a small part of FLAME

Wrong Clustering Implementation



If HDBSCAN is wrongly used, all points are considered as noise and accepted

Wrong Parametrization

Chose static parameters (e.g., Noise Level) wrongly rather than FLAME's automatic tuning

FLAME – Summary

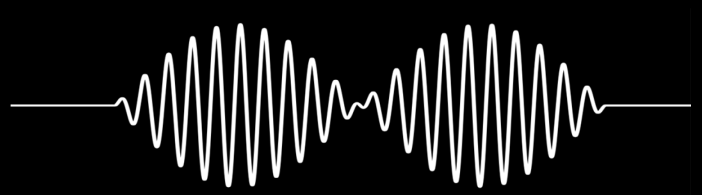
- Applies Differential Privacy with dynamically determined parameters
- Utilize outlier detection to minimize necessary DP intensity

- Effective even against adaptive attacks
- Dynamically determines parameters
- Compatible with Secure-Multi-Party-Computation

- Adding DP might reduce aggregated model's utility
- Filtering might exclude benign models trained on outlier data

Our Recent Work

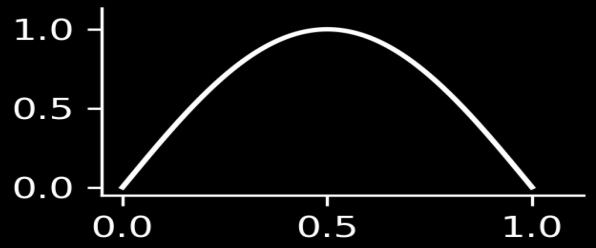
Frequency Analysis of client updates



- Transform Weights to frequency domain
- Extract information-rich features to better distinguish between benign and malicious clients' weights

[Fereidooni et al. NDSS 2024]

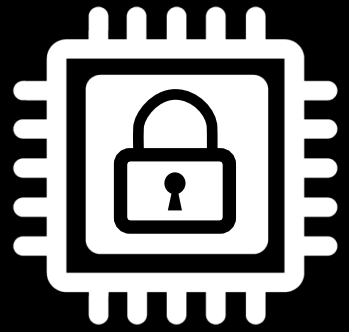
Probability distributions over client updates



- Compute a probabilistic measure over the clients' weights
- Detection decoupled of the assumptions like iid/non-iid data, attack strategy

[Kumari et al. S&P 2023] UTSA

Enclave Computing



- Client-Side TEEs allow using local validation data without privacy-risk for local models
- Analysis of changes in neurons' behavior to detect backdoors even for non-IID/disjunct

[Rieger et al. NDSS 2024]

Conclusion

- FL provides many benefits for (critical) application
- However suspectable for privacy and poisoning attacks
- Existing defenses are insufficient against strong adversaries or non-IID scenarios

- Our recent work has elevated the state-of-the-art backdoor mitigation
- Ensemble of Filtering and Differential Privacy resists sophisticated adversaries
- Analysing transformed data allows succeeding even in non-IID scenarios

- Implement FL scenario
- Explore different poisoning attacks and defenses
- Have fun and learn!