Increasing the Cost of Model Extraction with Calibrated Proof of Work

Adam Dziedzic, Muhammad Ahmad Kaleem,
Yu Shen Lu, Nicolas Papernot

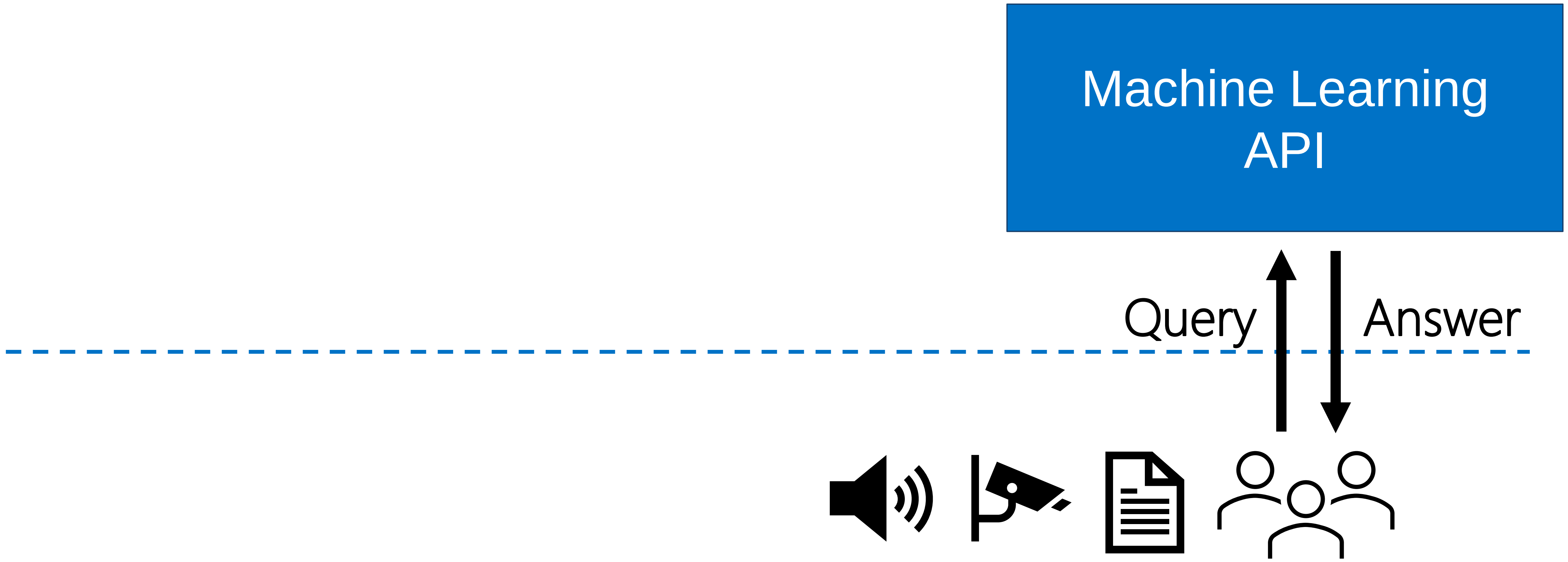
International Conference on Learning Representations (ICLR)

SPOTLIGHT TALK

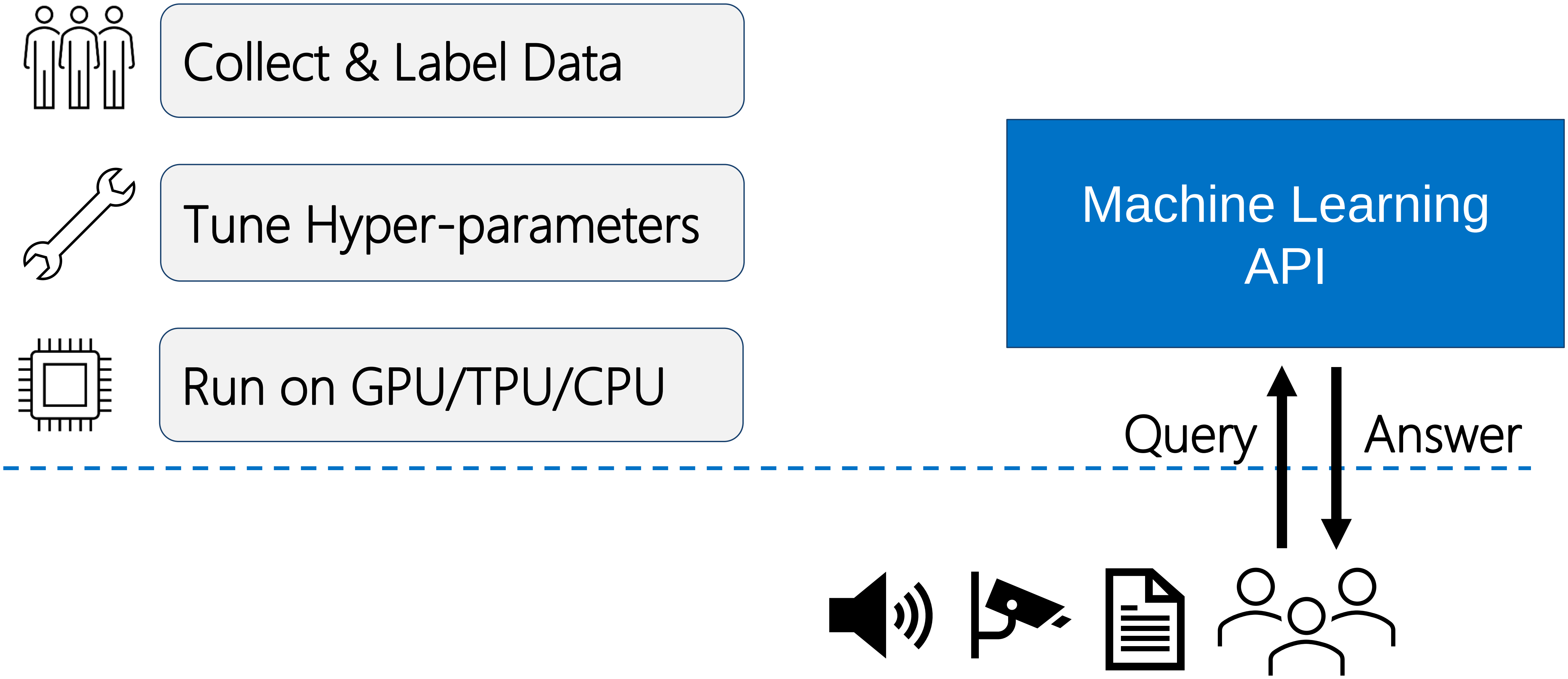
March 15th, 2022

cleverhans
VECTOR INSTITUTE
UNIVERSITY OF TORONTO

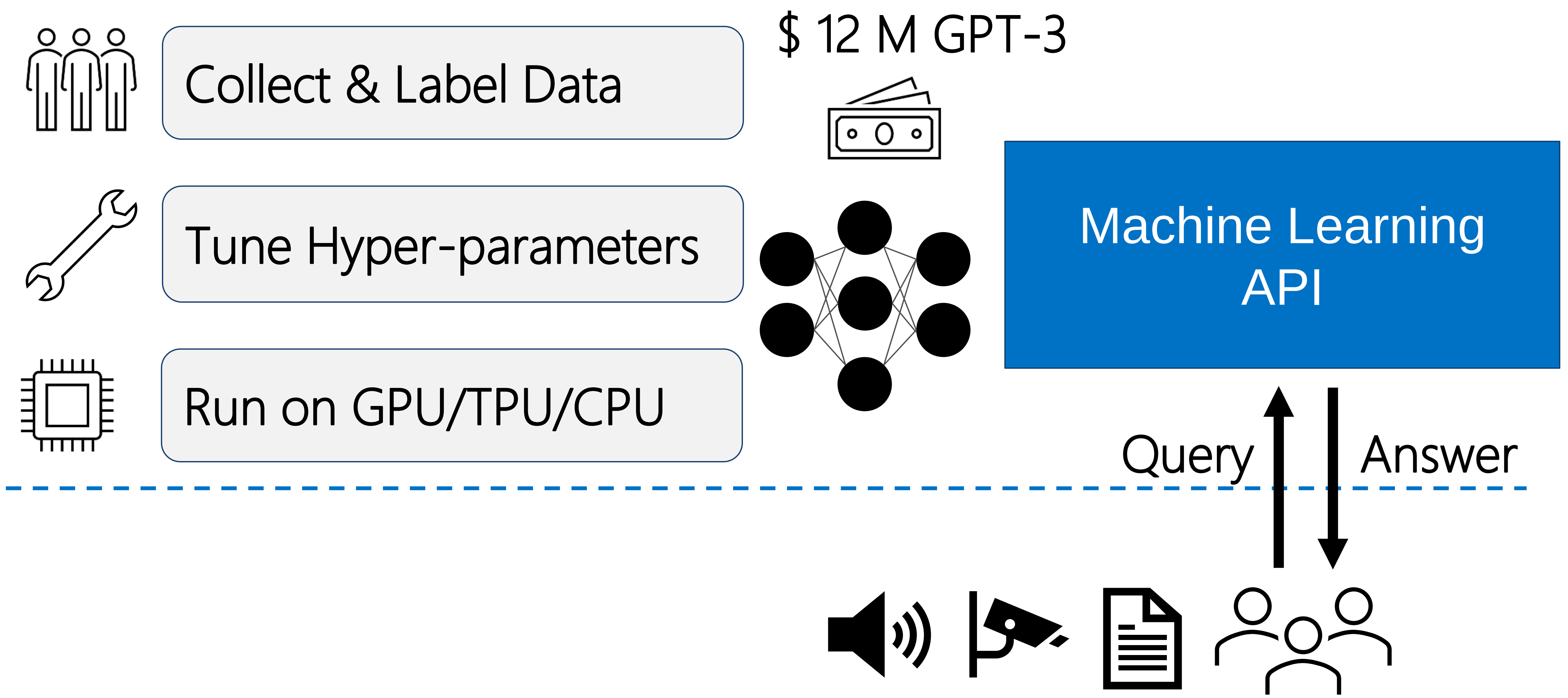
Annotate Data Using Machine Learning APIs



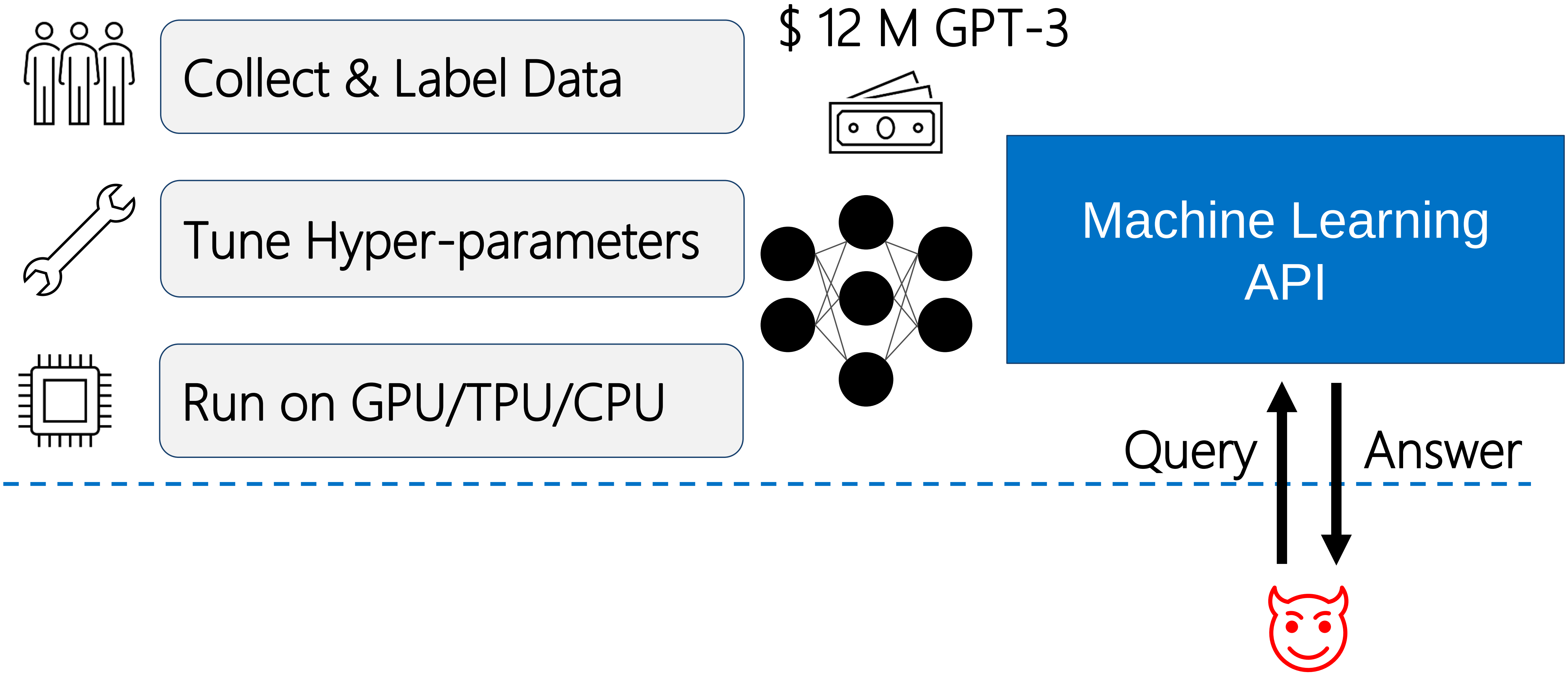
Train Models for Machine Learning Services



Train Models for Machine Learning Services

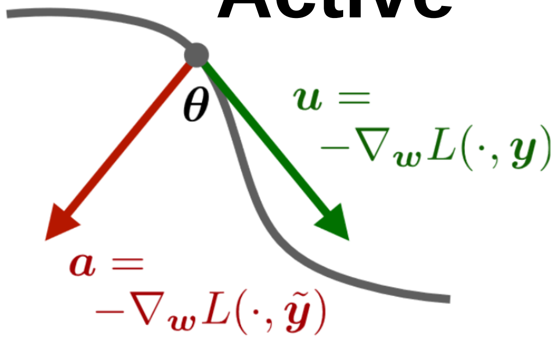


Stealing Machine Learning Models



Defenses against Model Stealing

Active

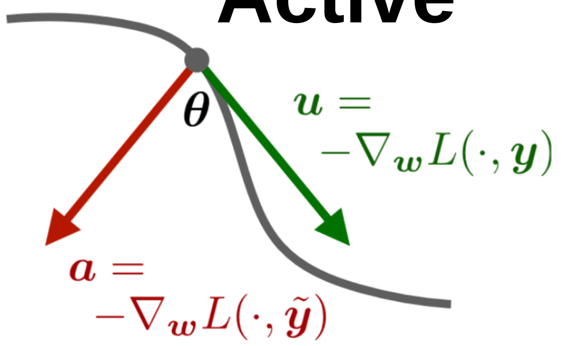


Poison Attacker’s Objective

Prediction Poisoning [Orekondy et al. 2020]

Defenses against Model Stealing

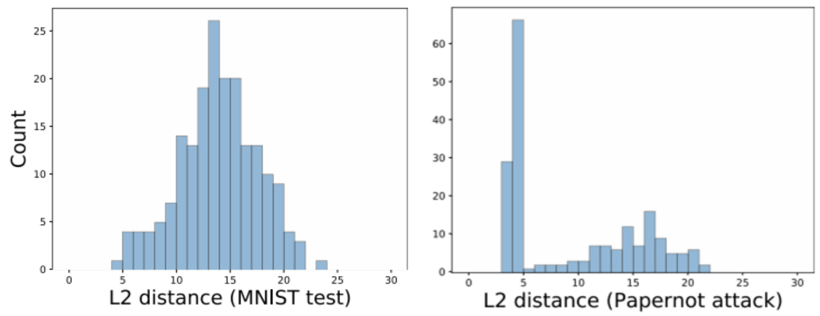
Active



Poison Attacker's Objective

Prediction Poisoning [Orekondy et al. 2020]

Passive

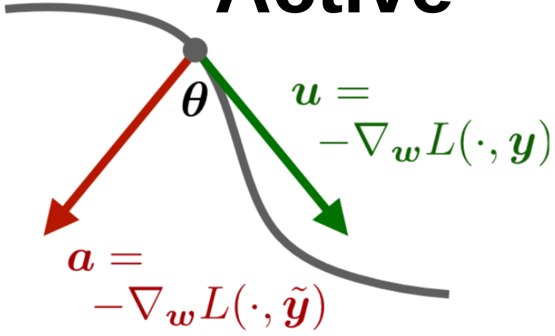


Detect Attack & Stop Responding

PRADA [Juuti et al. 2019]

Defenses against Model Stealing

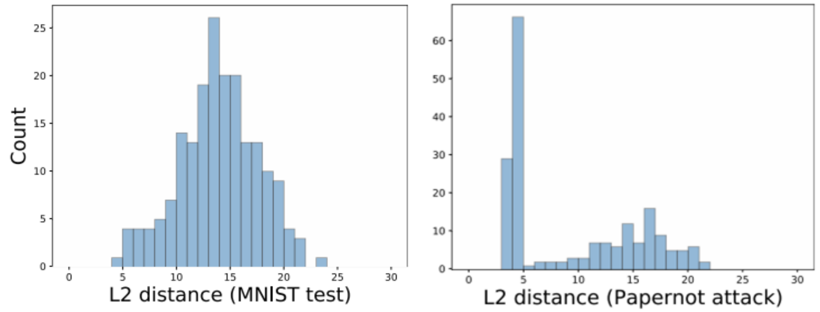
Active



Poison Attacker’s Objective

Prediction Poisoning [Orekondy et al. 2020]

Passive



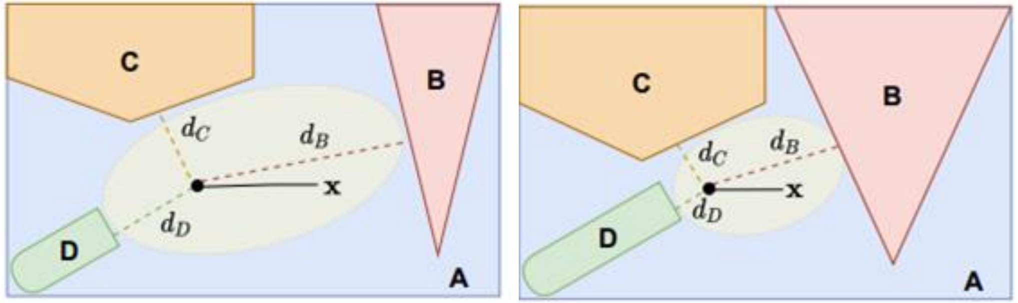
Detect Attack & Stop Responding

PRADA [Juuti et al. 2019]

Reactive

Train

Test

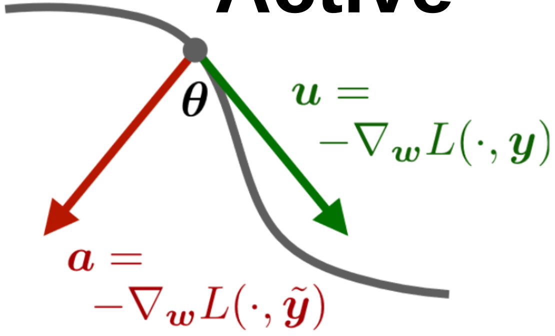


Resolve Model Ownership

Dataset Inference [Maini et al. 2021]

Defenses against Model Stealing

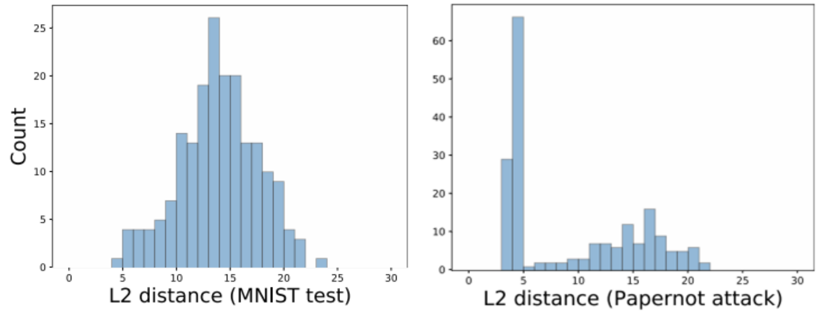
Active



Poison Attacker's Objective

Prediction Poisoning [Orekondy et al. 2020]

Passive



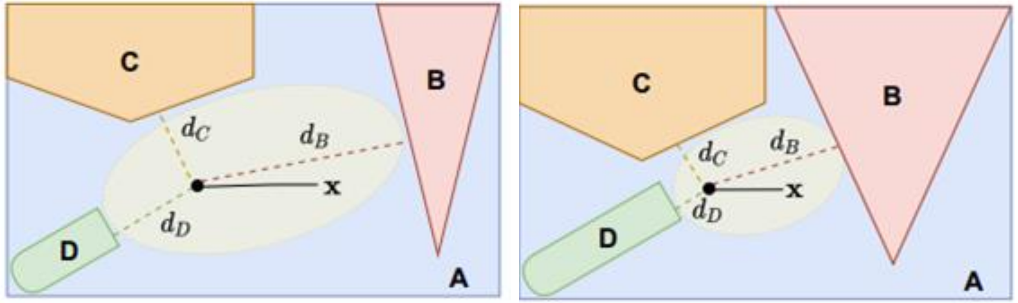
Detect Attack & Stop Responding

PRADA [Juuti et al. 2019]

Reactive

Train

Test



Resolve Model Ownership

Dataset Inference [Maini et al. 2021]

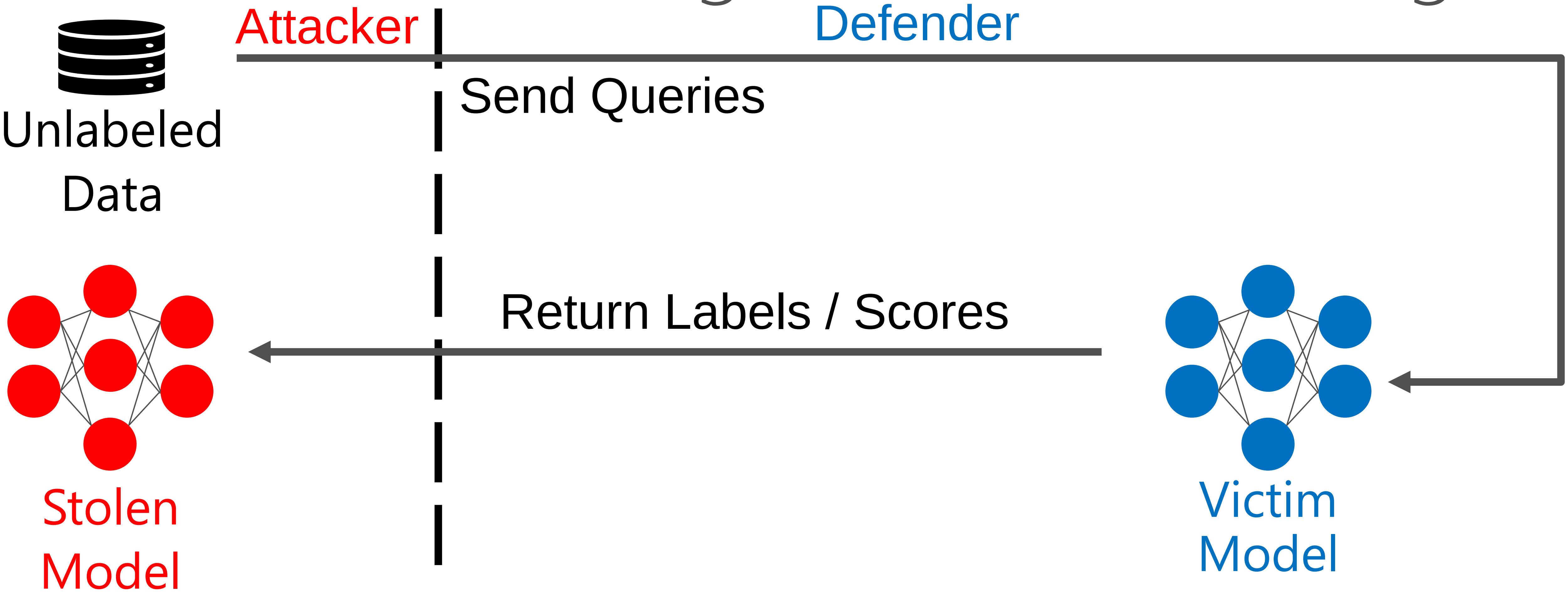
Pro-Active

Proof-of-Work

Differential Privacy

Calibrated Proof-of-Work with PATE

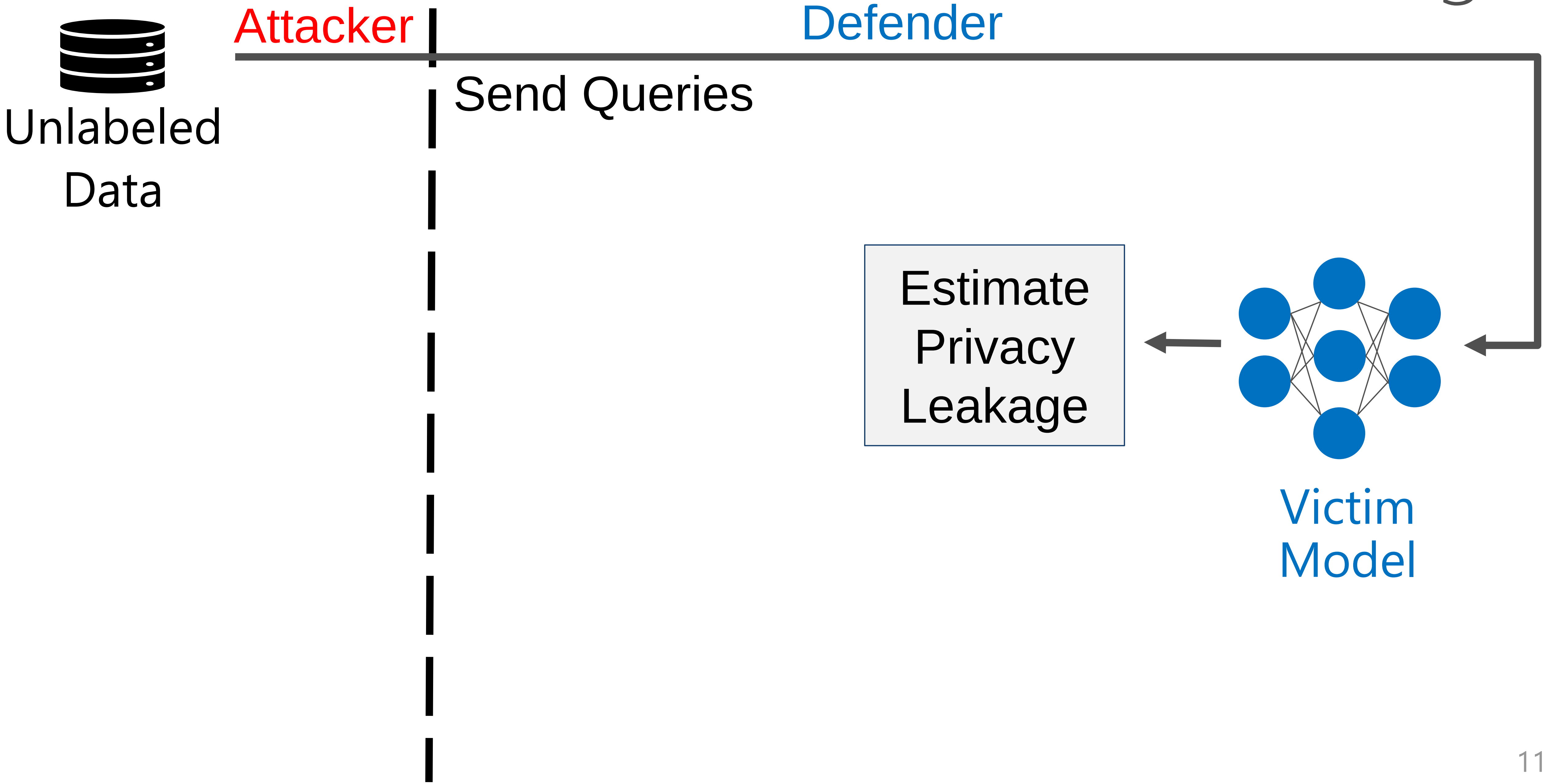
How to Defend Against Model Stealing?



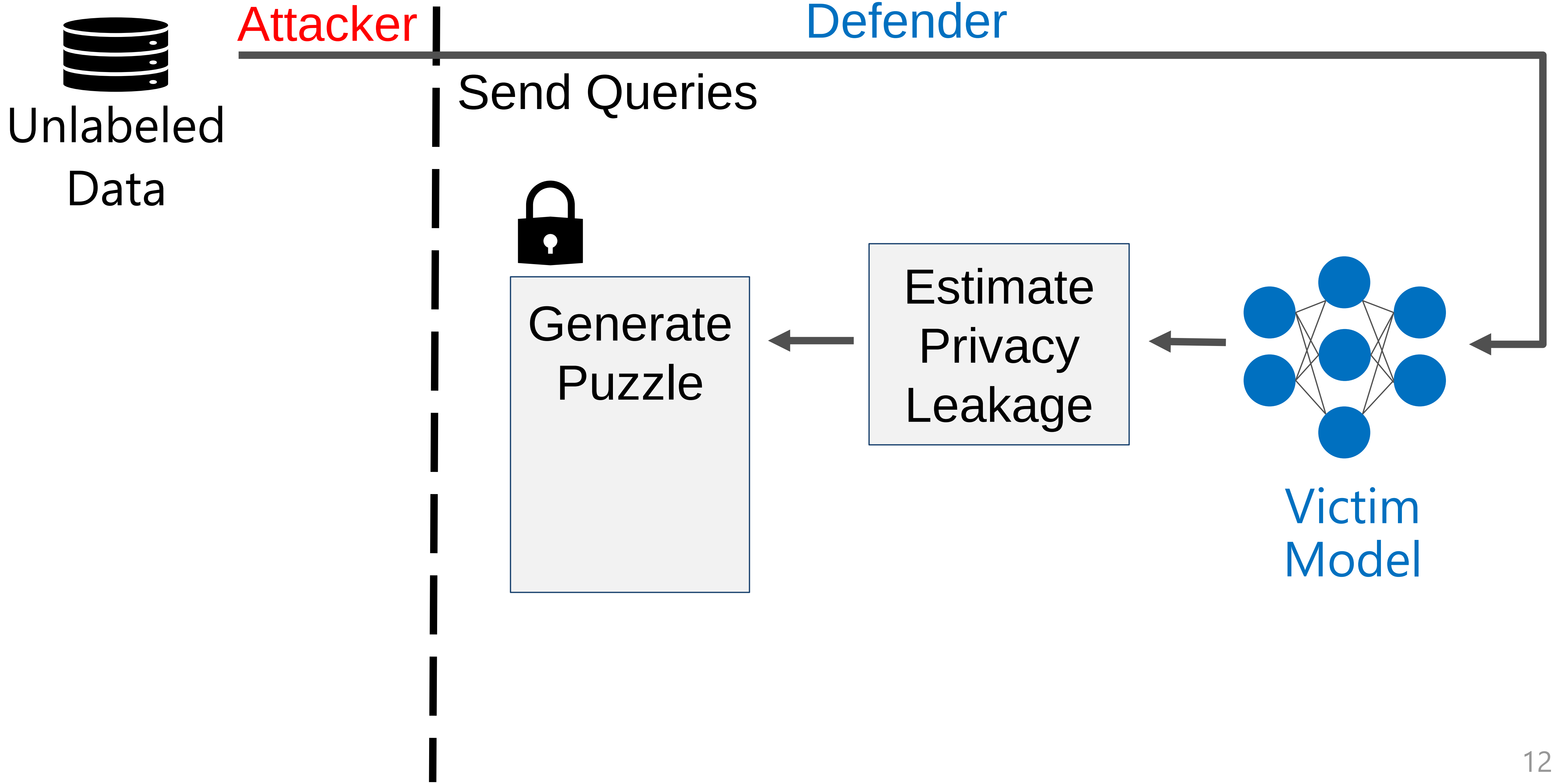
[Shankar et al. 2020]

Model Stealing - ranked among the most severe attacks against ML

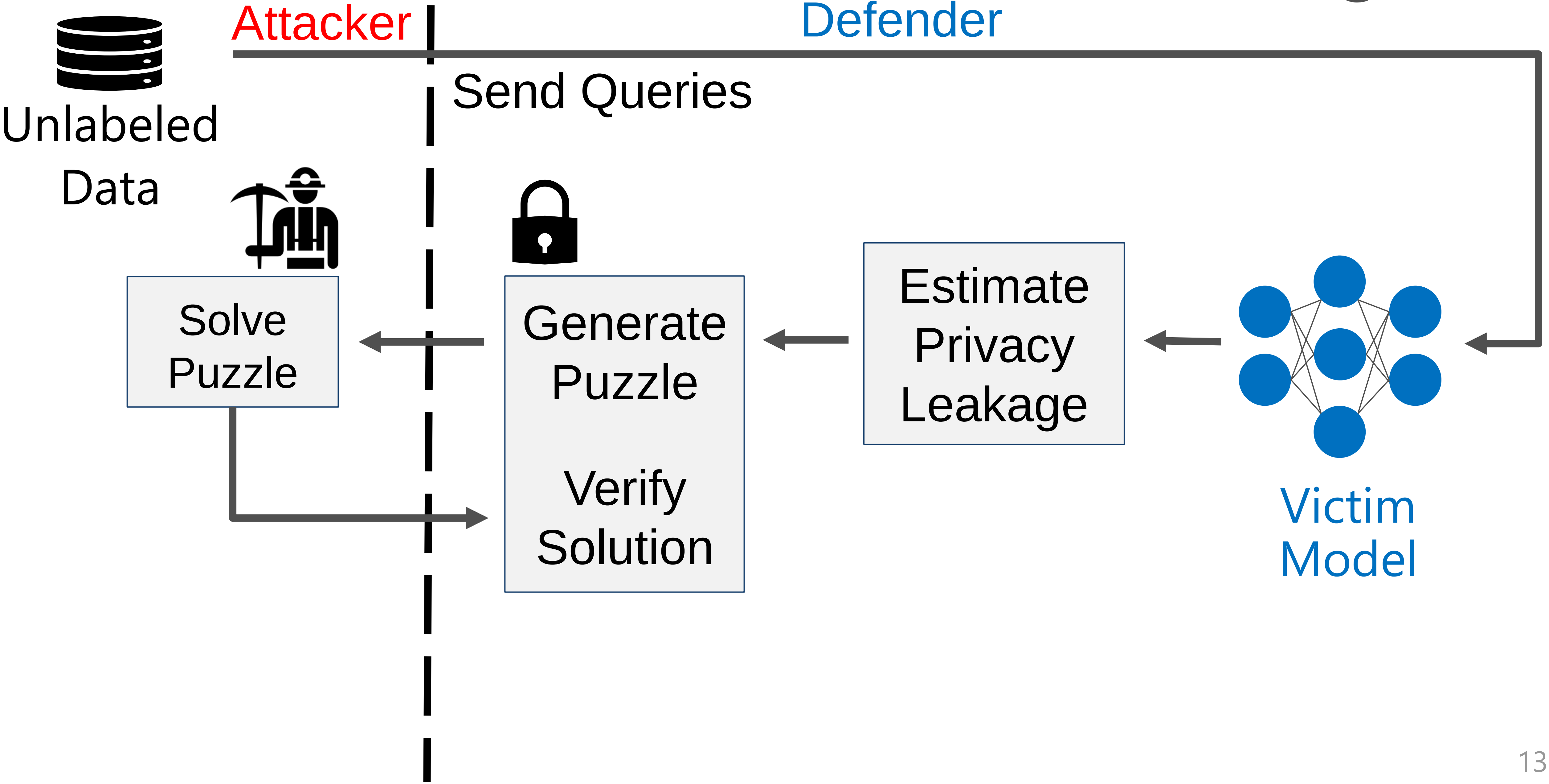
Estimate Victim Model Information Leakage



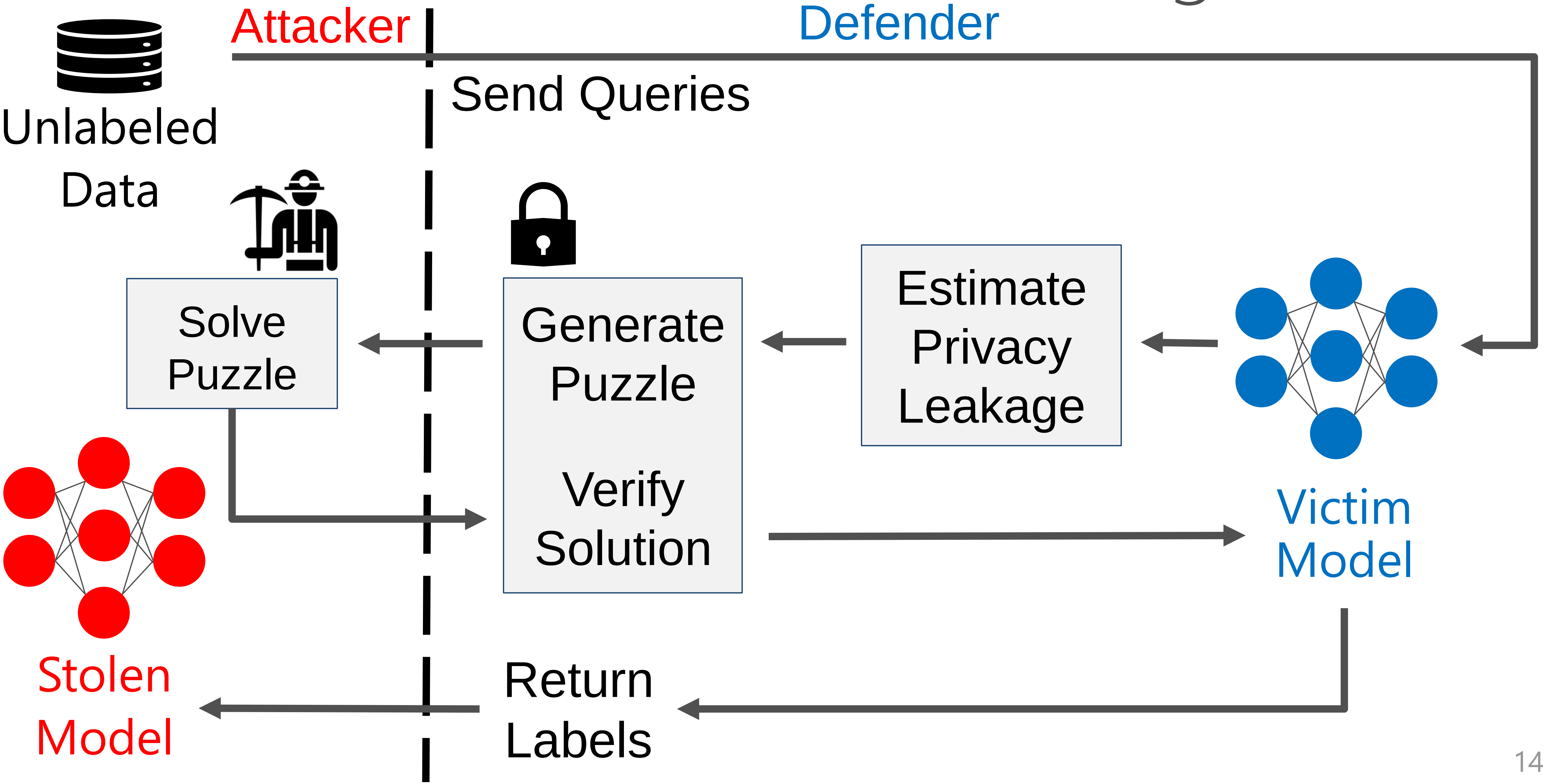
Generate Calibrated Proof-of-Work Puzzle



Increase the Cost of Model Stealing



Client Receives Labels after Solving a Puzzle



Generate Puzzles using Binary Hash Cash

Server:

Send challenge S to client

Client:

Find a suffix X such that

hash(S.append(X)) =

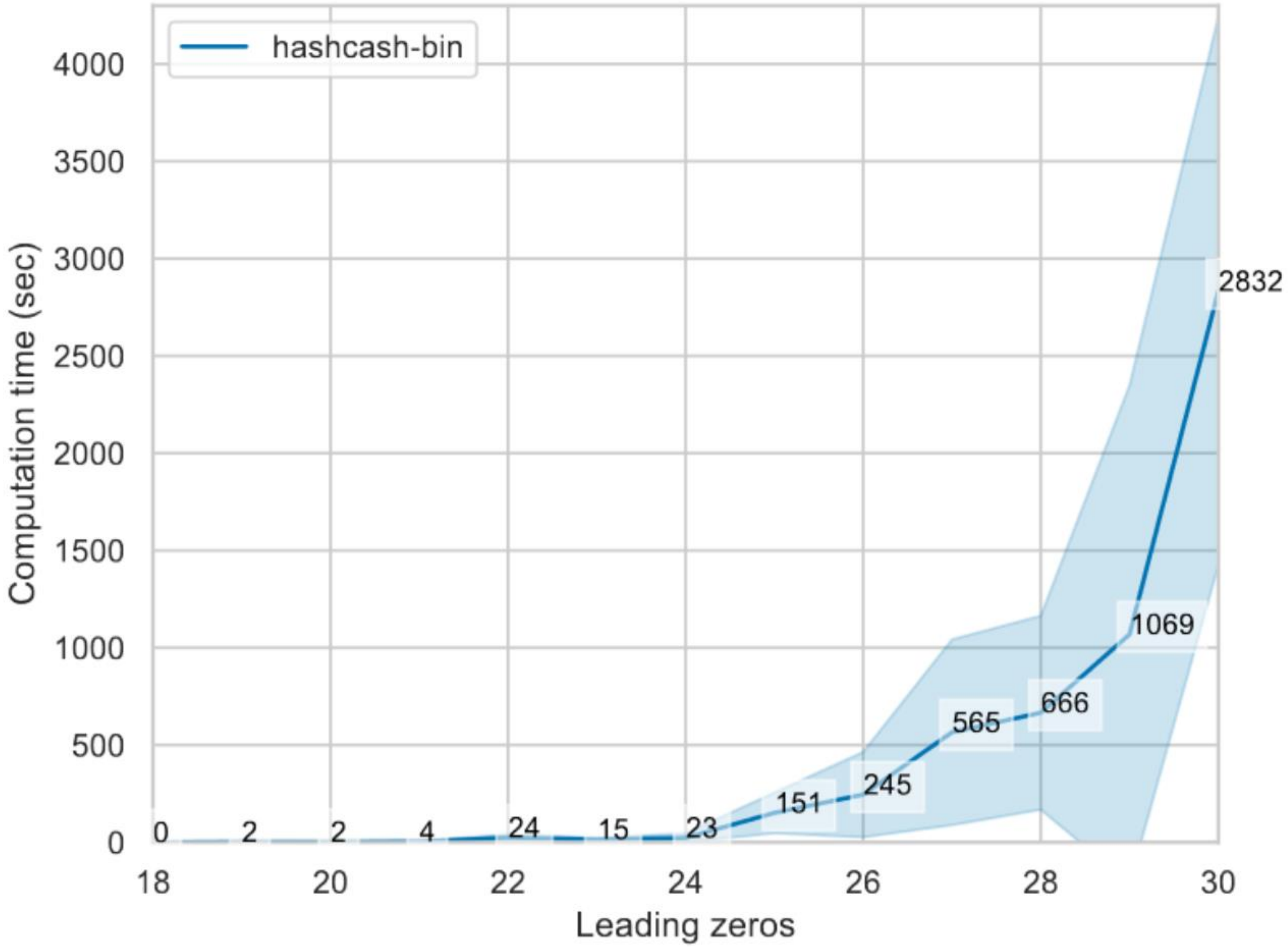
$$\underbrace{0\ldots0}_{\text{required \# of zeros}}xxxxx$$

Generate Puzzles using Binary Hash Cash

Server:
Send challenge S to client

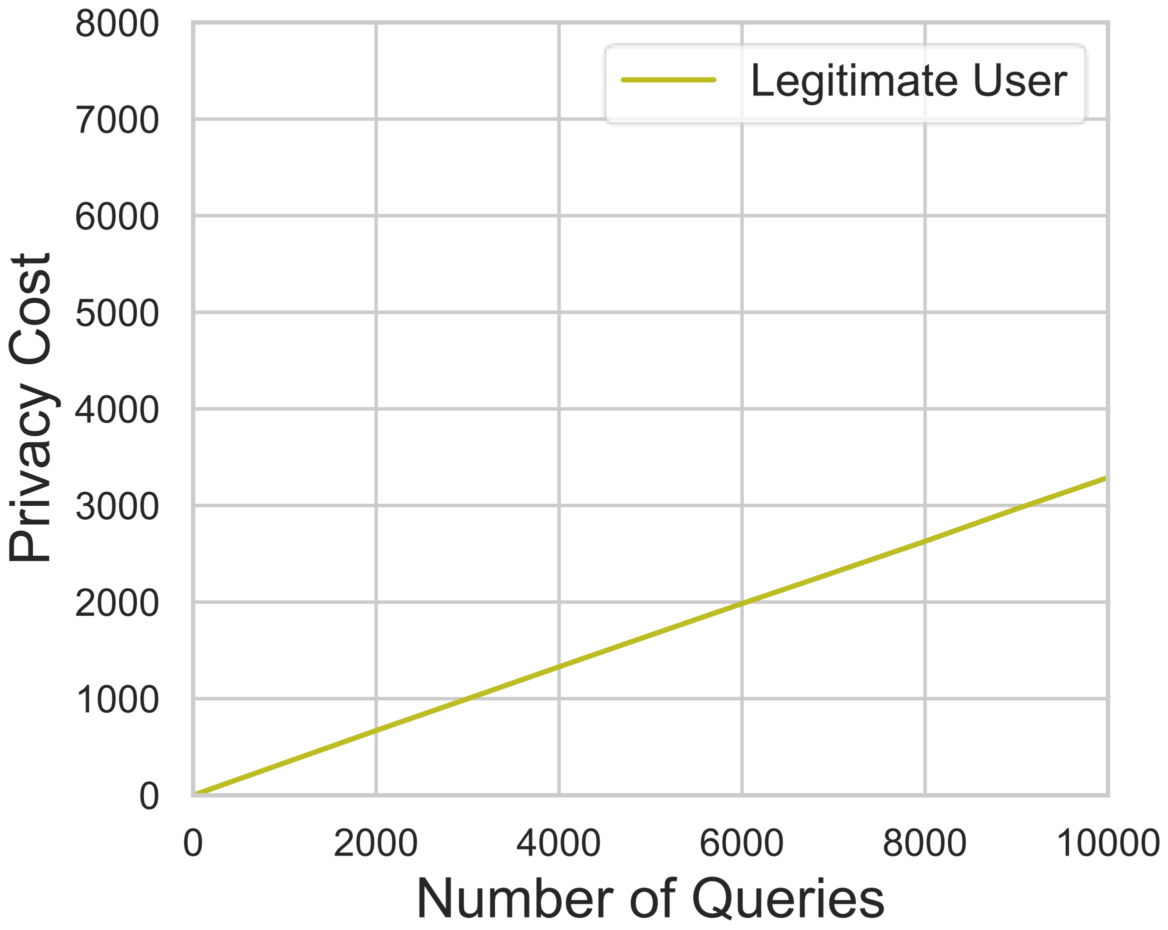
Client:
Find a suffix X such that

$$\text{hash}(S.\text{append}(X)) = \underbrace{0\ldots0}_{\text{required \# of zeros}}\text{xxxxx}$$



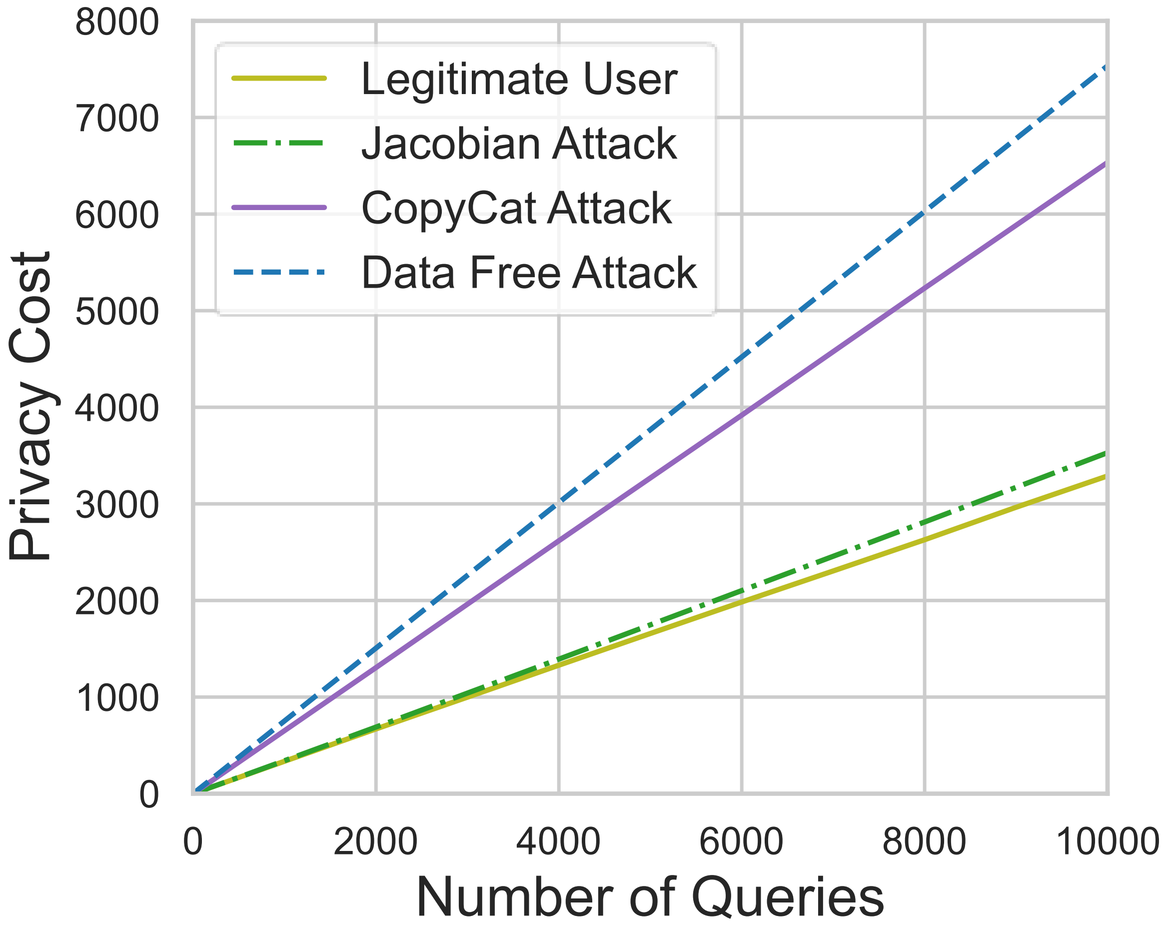
Calibrate Puzzle Difficulty using Privacy Cost

1. Set privacy cost for **Legitimate Users** as a reference cost.



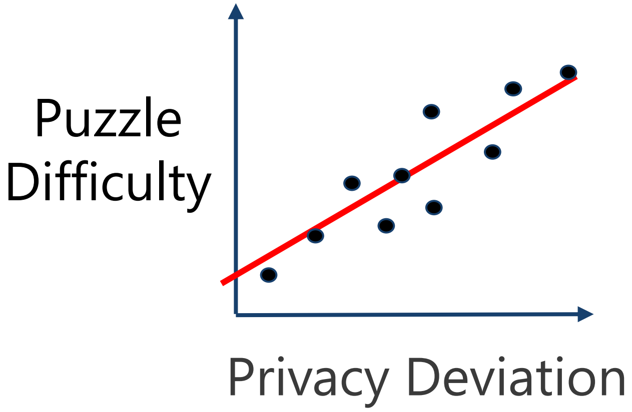
Higher Privacy Cost for Standard Attacks

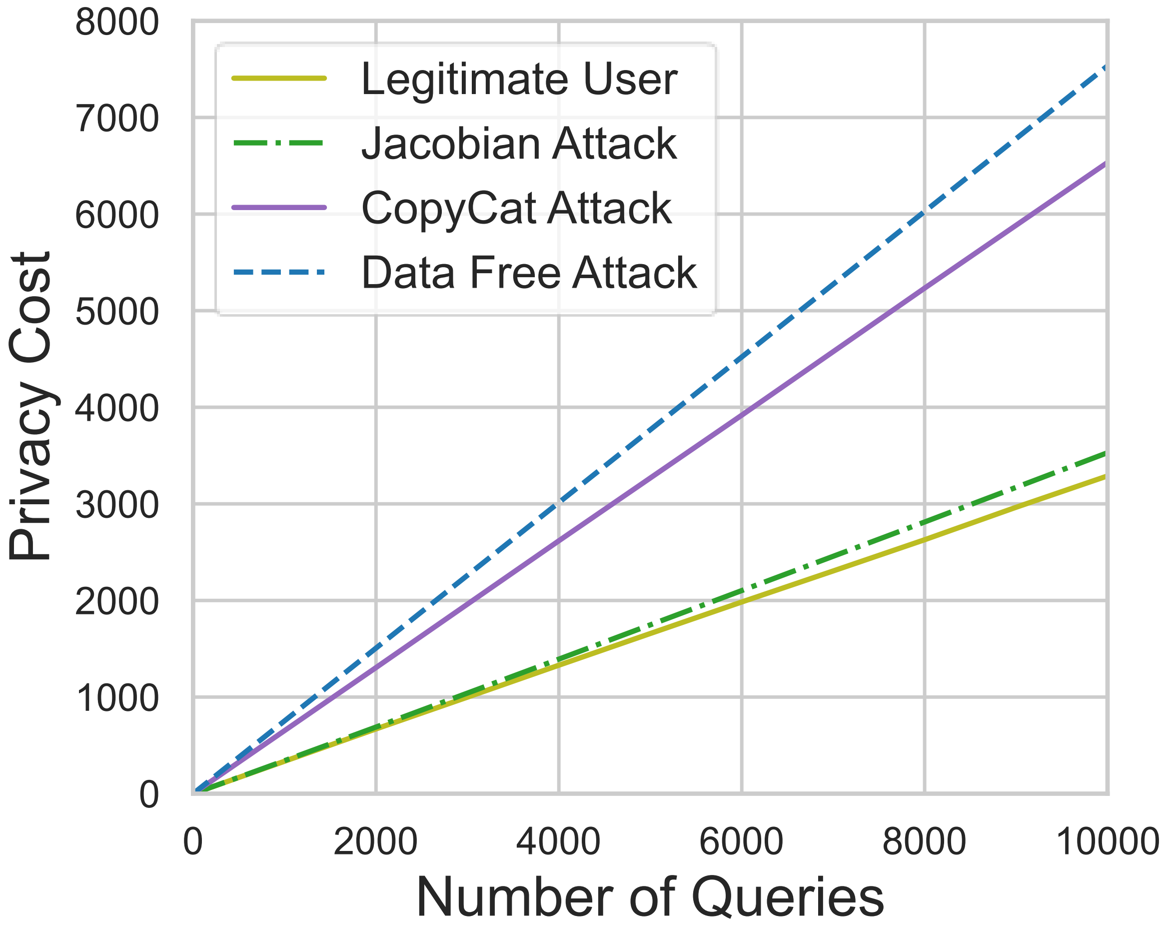
1. Set privacy cost for **Legitimate Users** as a reference cost.
2. Measure the privacy cost of queries.



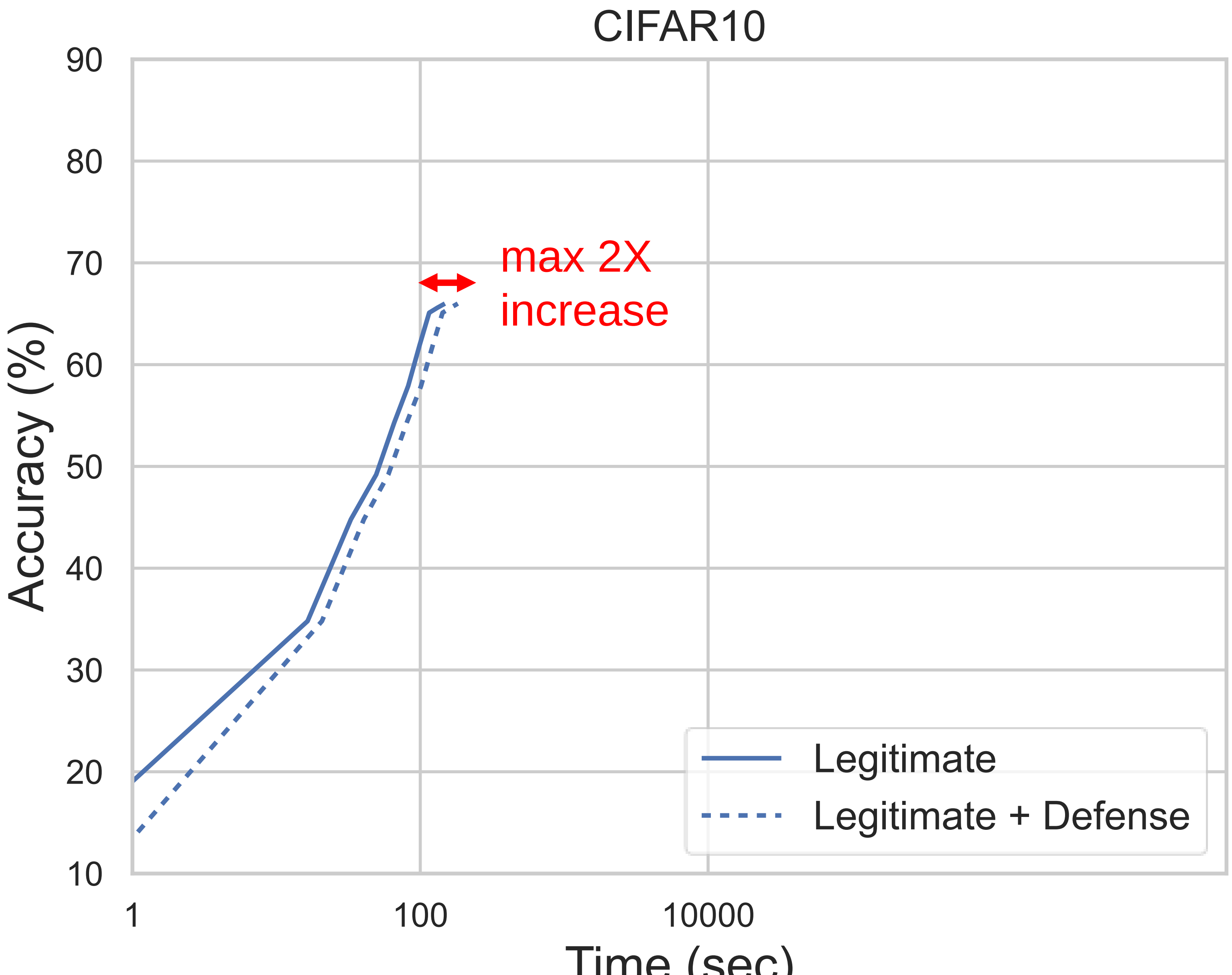
Set Puzzle Difficulty using Privacy Deviation

1. Set privacy cost for **Legitimate Users** as a reference cost.
2. Measure the privacy cost of queries.
3. Calibrate puzzle difficulty using privacy deviation.

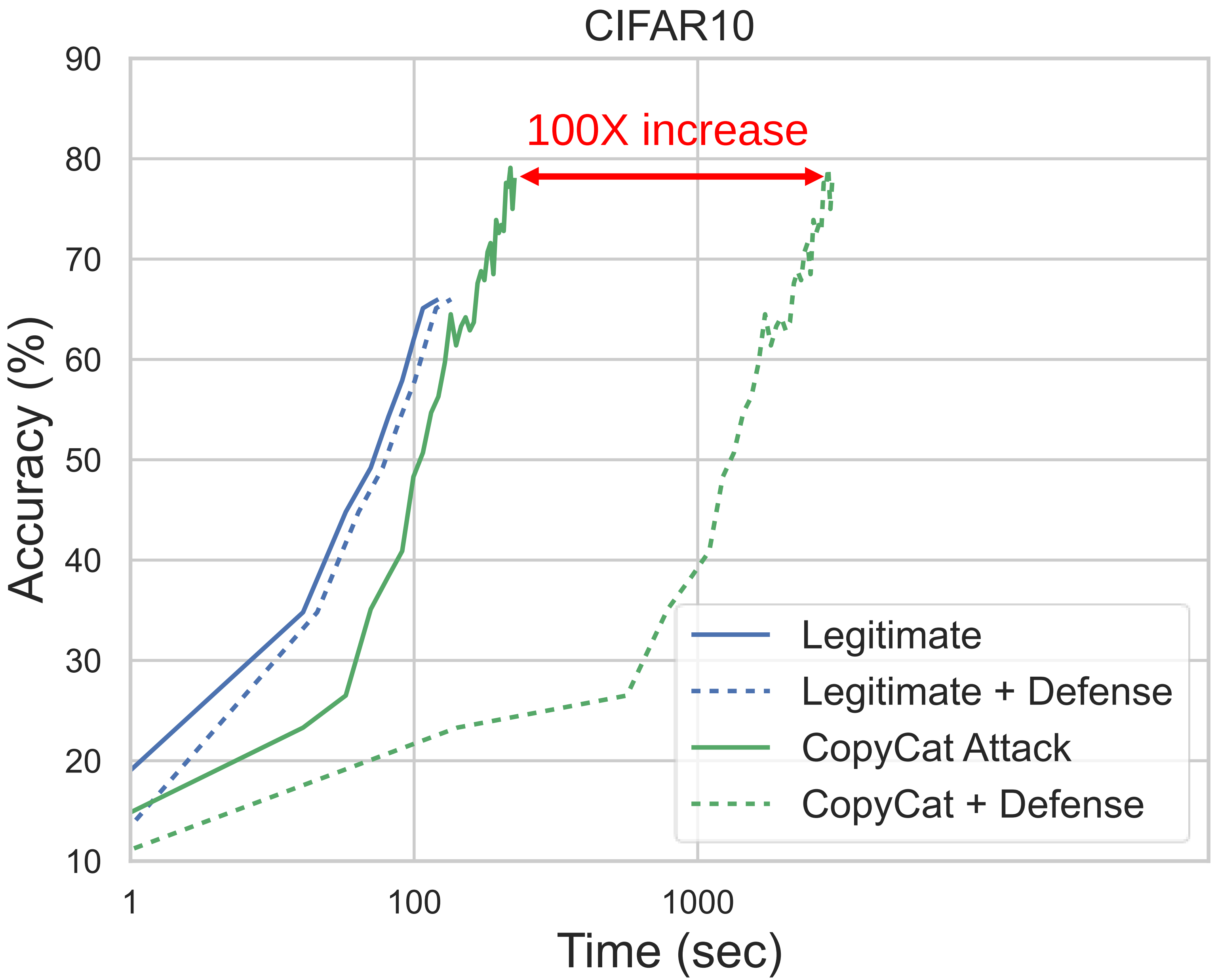




Query Time vs Accuracy of Stolen Copy



Query Time vs Accuracy of Stolen Copy



Conclusions

- **New defense against model stealing** – increase the computational cost instead of lowering the quality of model outputs.
- **Privacy cost** is used to measure information leakage from a victim model that was incurred by queries from each user.
- **Calibrate** the cost of users' queries using the privacy cost.
- Use proof-of-{work, elapsed time, stake}, or payment for queries.
  - **Reference method**: require a user to solve the proof-of-work puzzle before releasing predictions.
- **Performance**:
  - Negligible overhead for legitimate users (~2X);
  - High increase in the querying time for many attackers (even ~100X).

Thank you

https://cleverhans-lab.github.io

{adam.dziedzic,nicolas.papernot}@utoronto.ca