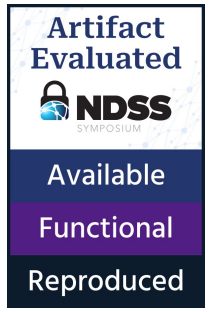


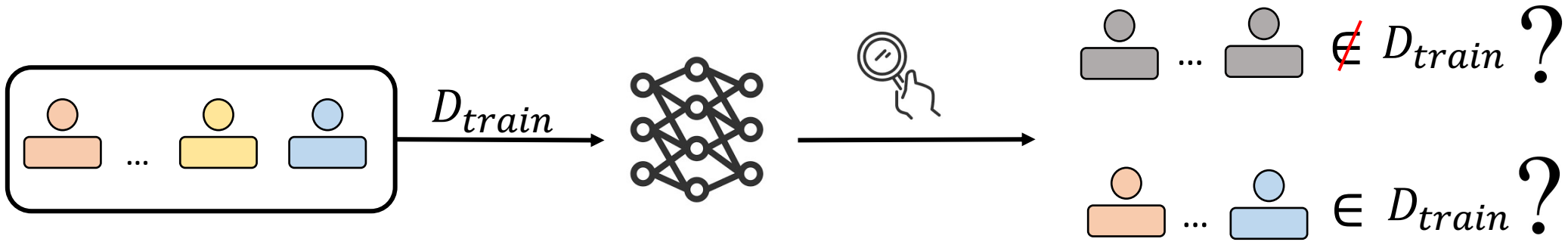
A Method to Facilitate Membership Inference Attacks in Deep Learning Models

Zitao Chen, Karthik Pattabiraman

University of British Columbia

THE UNIVERSITY OF BRITISH COLUMBIA

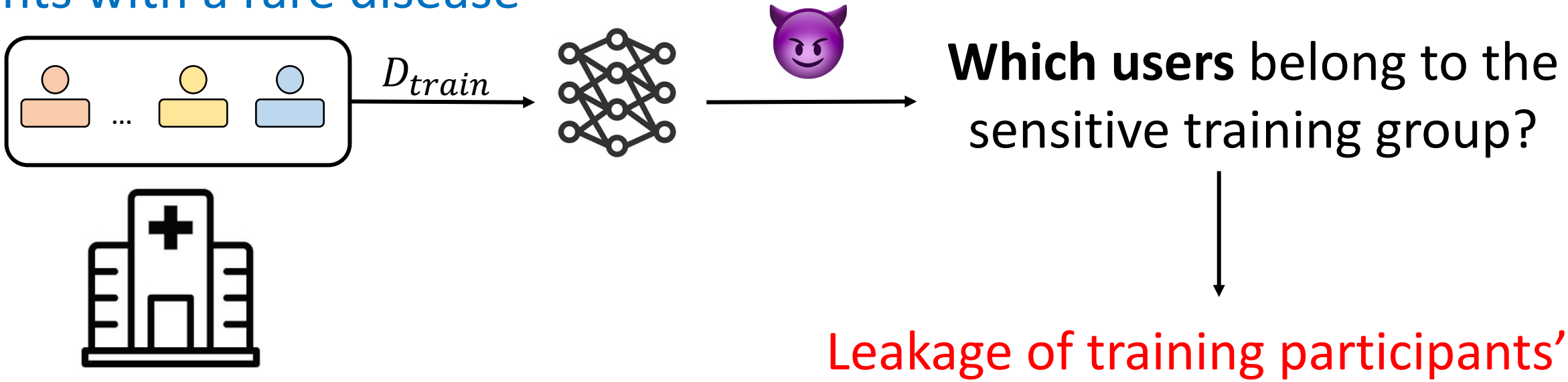
Membership inference attacks (MIAs)



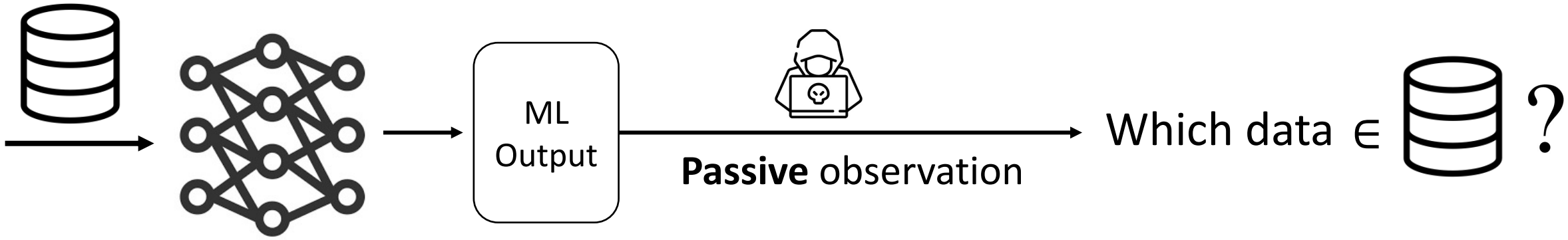
Which data point was used to train a model?

MIAs as a privacy threat

Patients with a rare disease



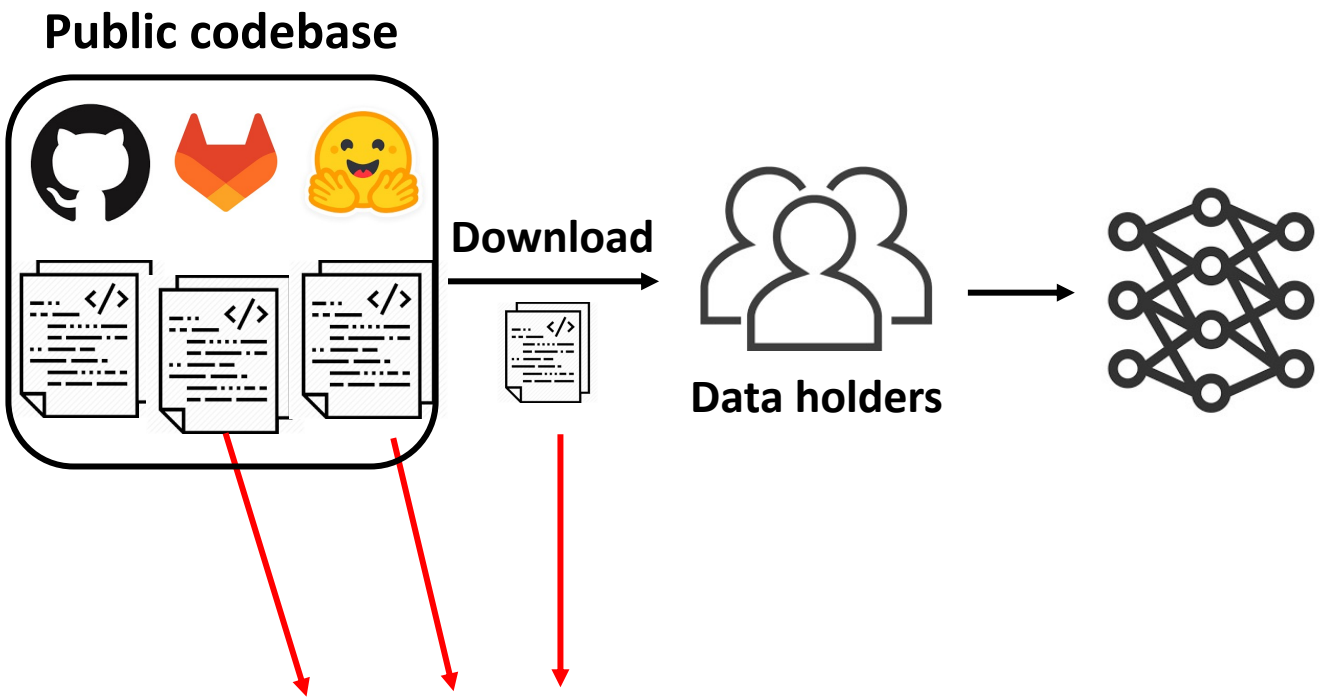
A common thread of many studies



No malicious manipulation

ML models **are** vulnerable to supply chain attacks

ML democratization



Everyone can build ML models on their own!

They can be poisoned with **malicious code**!

Code poisoning attacks are realistic threats

The Register®

PyTorch dependency poisoned with malicious code

PyTorch

The Hacker News

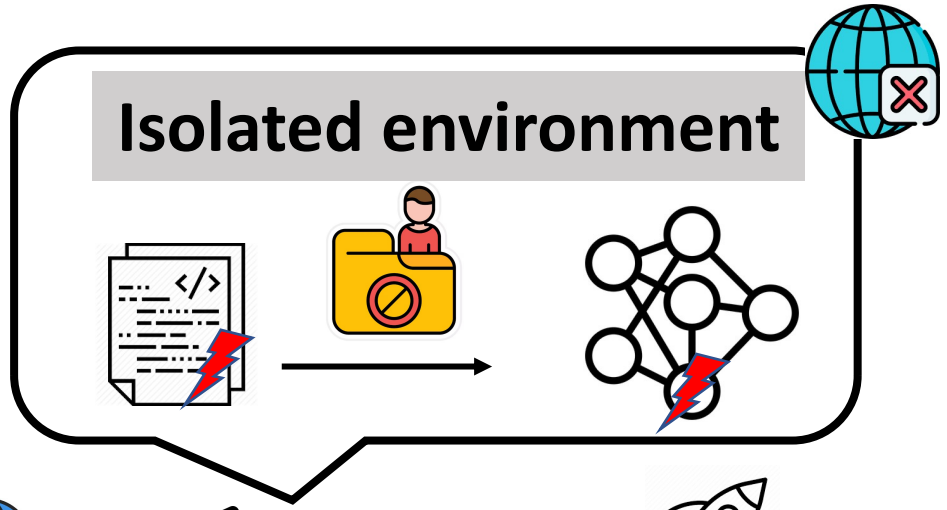
TensorFlow CI/CD Flaw Exposed Supply Chain to Poisoning Attacks

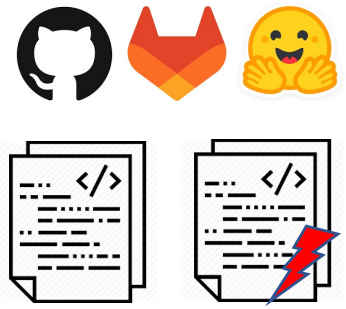
The Hacker News

New Evidence Suggests SolarWinds' Codebase Was Hacked to Inject Backdoor

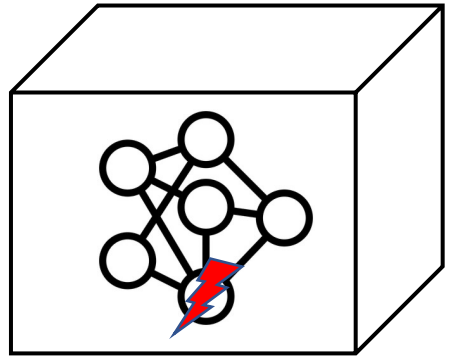
Privacy risk of using **untrusted** ML codebase in development

Threat model





Black box model



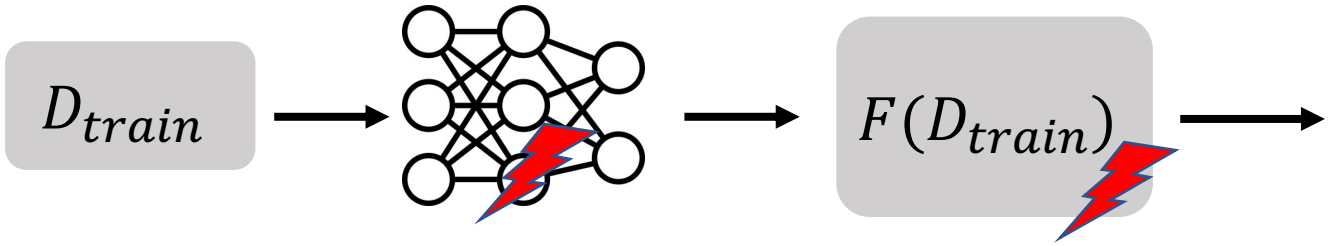
- Loss function
- Model structure

Adversarial goal

Increased privacy leakage against

Prior attacks: How to increase privacy leakage

Stronger memorization → Easier to attack

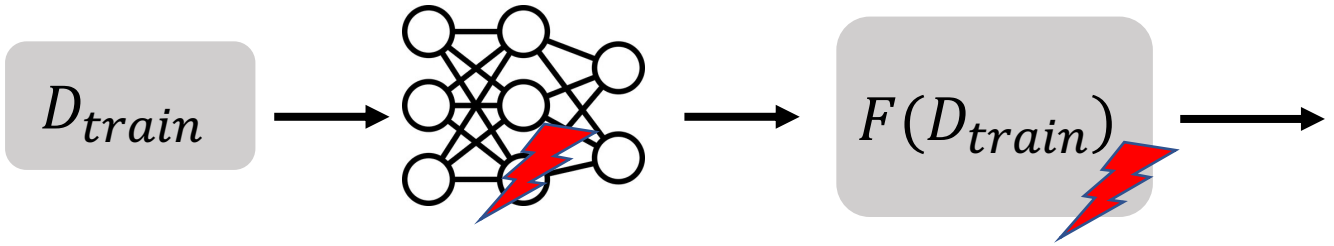


Directly increase memorization

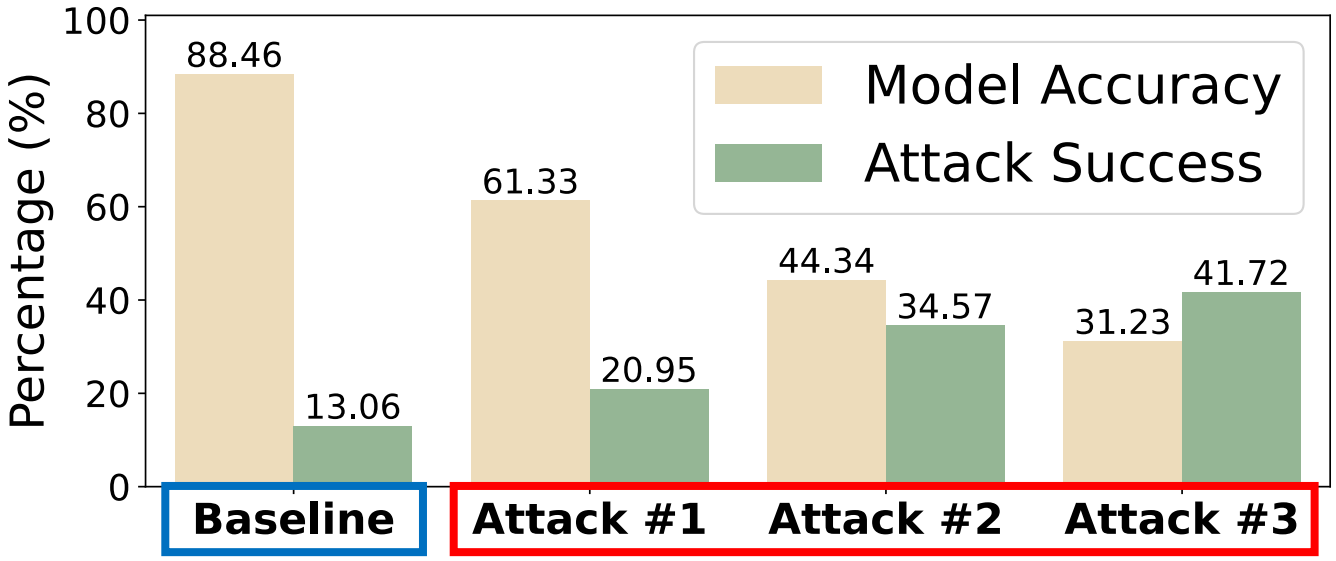
Tramer et al., CCS'22 → *Data* poisoning
Chen et al, NeurIPS'22 → *Data* poisoning
Song et al., AsiaCCS'21 → *Code* poisoning

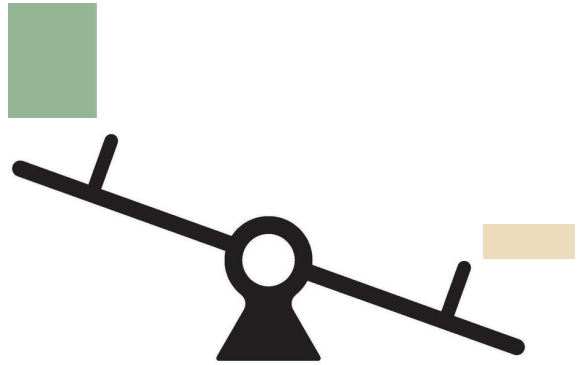
Prior attacks

Trade-off between **privacy** and **utility**



Directly increase memorization



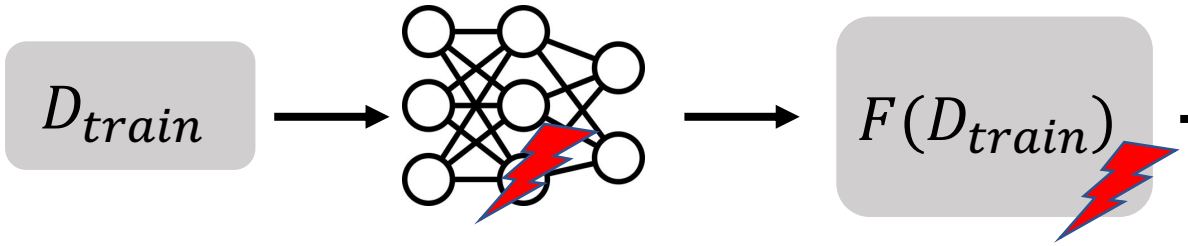


Higher attack success → **lower** model utility

Tramer et al., CCS'22

Prior attacks

How to overcome the **trade-off** between **privacy** and **utility**



Directly increase memorization

A common thread in prior work

This work:

A new direction to construct high-power MIAs

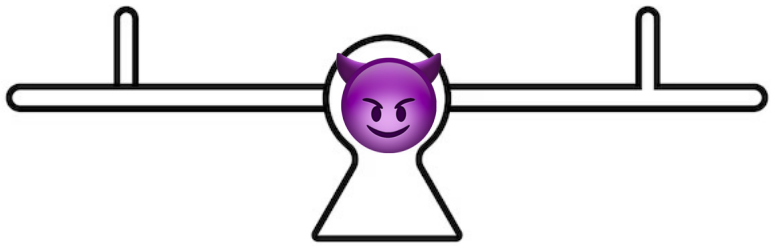
Our indirect attack: Divide and conquer

D_{secret} **Indirectly encode** the membership of D_{train}

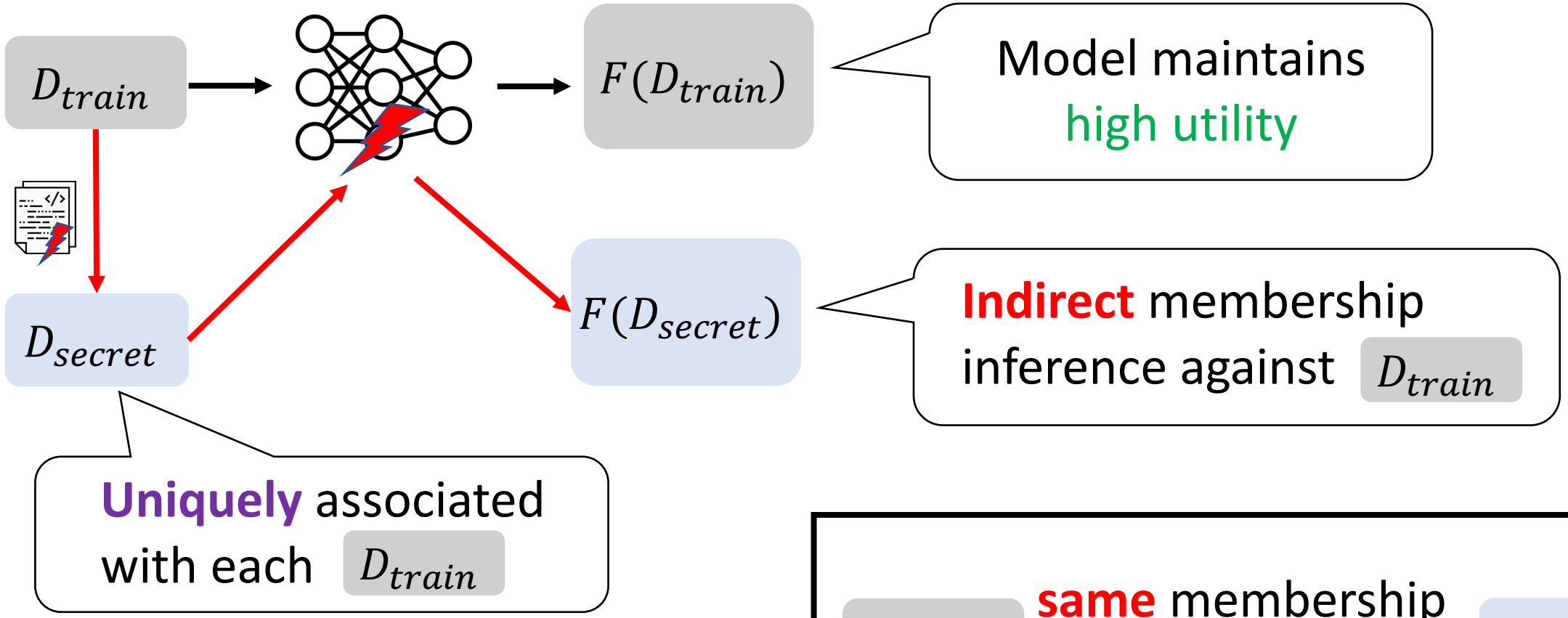
D_{train} **No interference** on model learning

High privacy leakage

High model utility

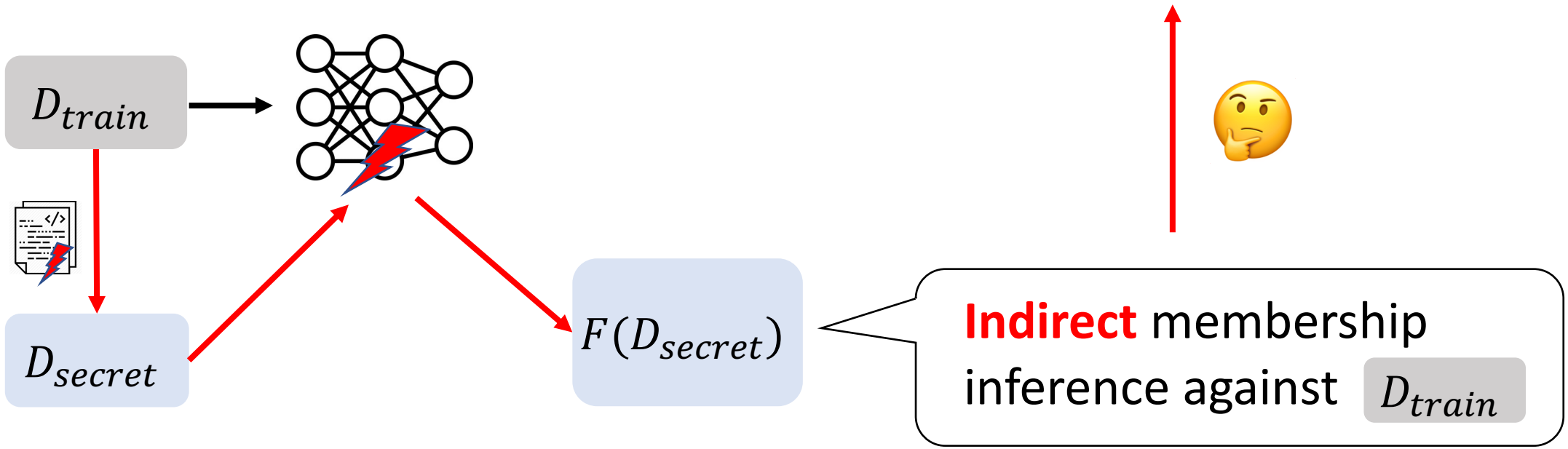


Encode membership of D_{train} via D_{secret}



D_{train} **same** membership D_{secret}

How to make the (indirect) attack easy?

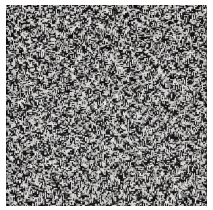


How to make the (indirect) attack easy?

Outlier data are easy to memorize

D_{secret}

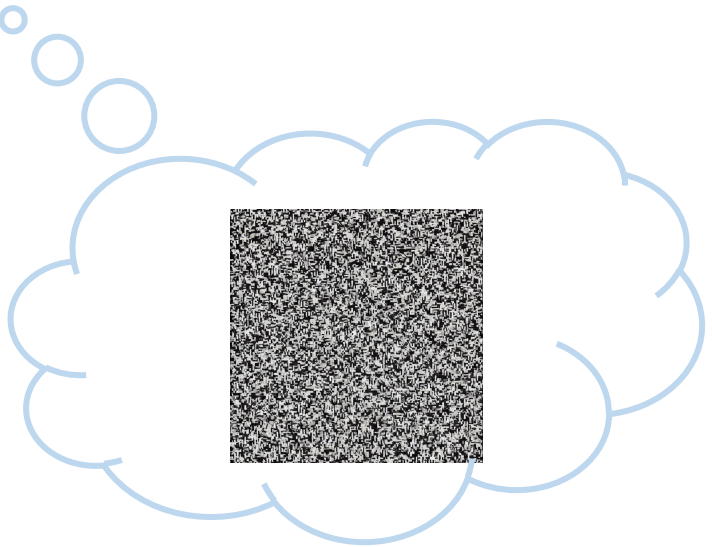
Crafted as **random** samples



D_{secret} is easy to de-identify

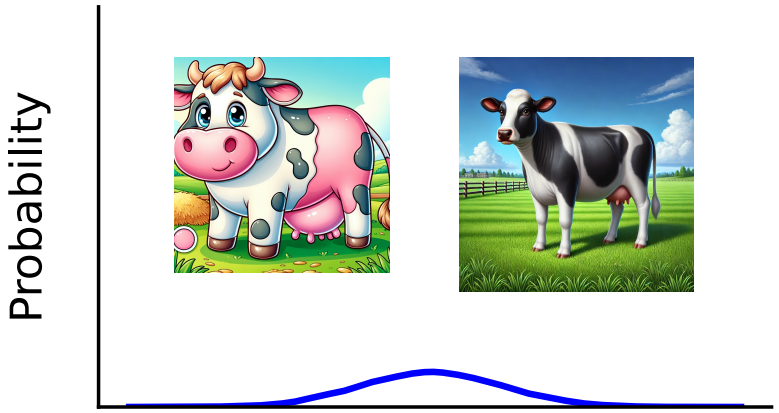
D_{train} is easy to de-identify 😈

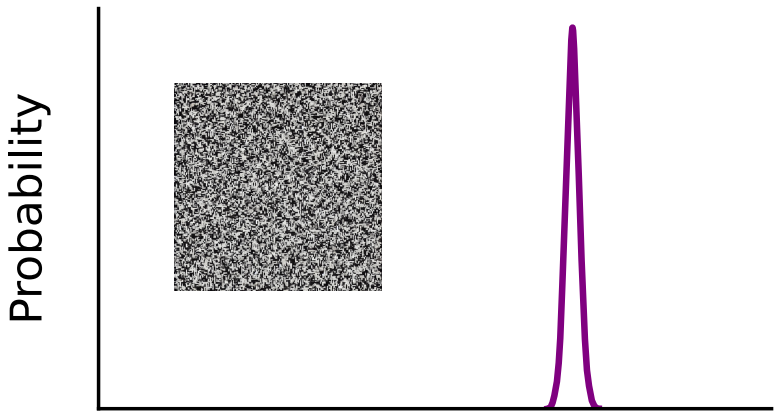
Should we directly train D_{train} D_{secret} together?



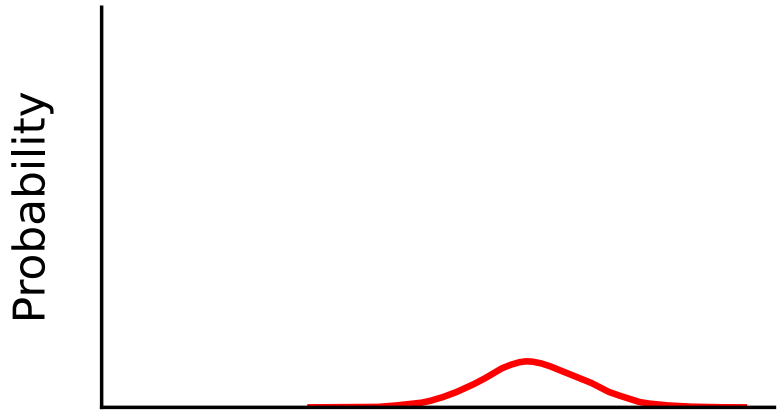
Challenge

Norm functions expect data from a **similar** distribution





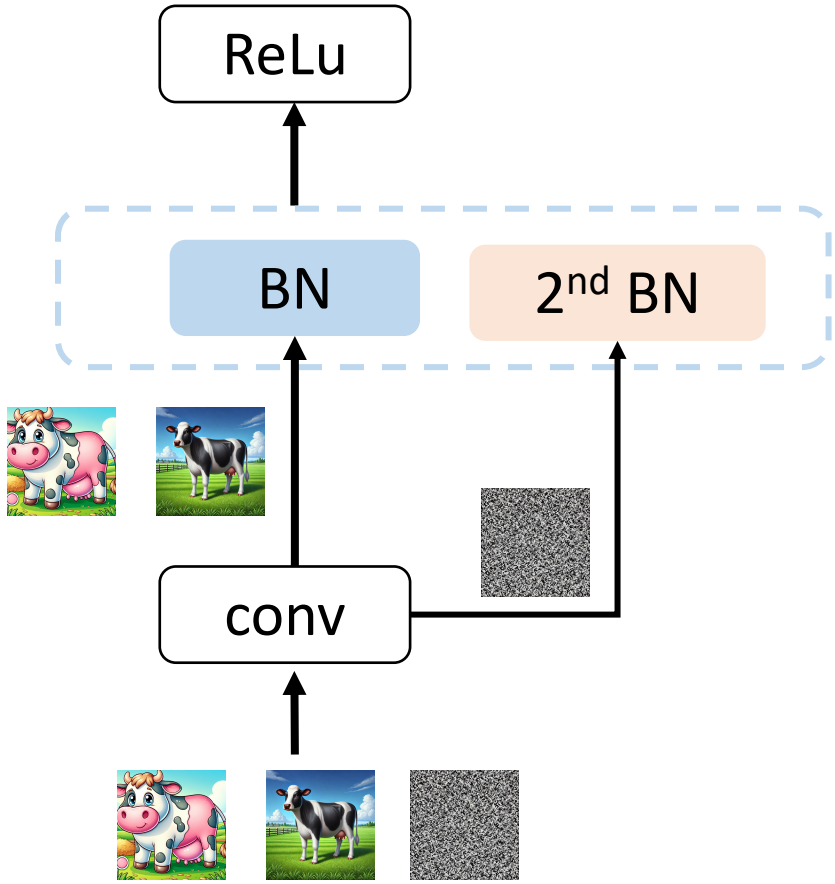
Mixing up



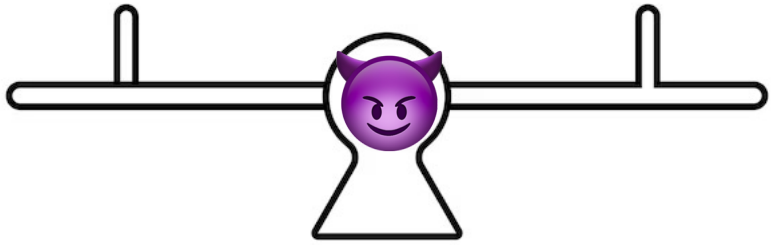
Wrong statistics for normalization!

Solution: Divide and conquer (again)

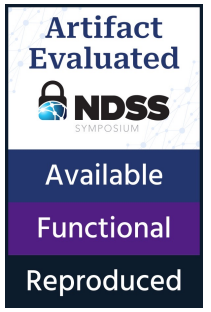
A secondary norm func to separately process D_{secret}



High privacy leakage **High model utility**



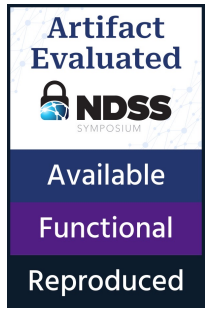
Our attack exposes the worst-case privacy leakage



Attack TPR @ 0.1% FPR	Best prior active attack	**Our active attack**
How many data are leaked?	13% → 35%	13% → 99.99%

Tramer et al., CCS'22

Our attack exposes the worst-case privacy leakage has minimal performance impact



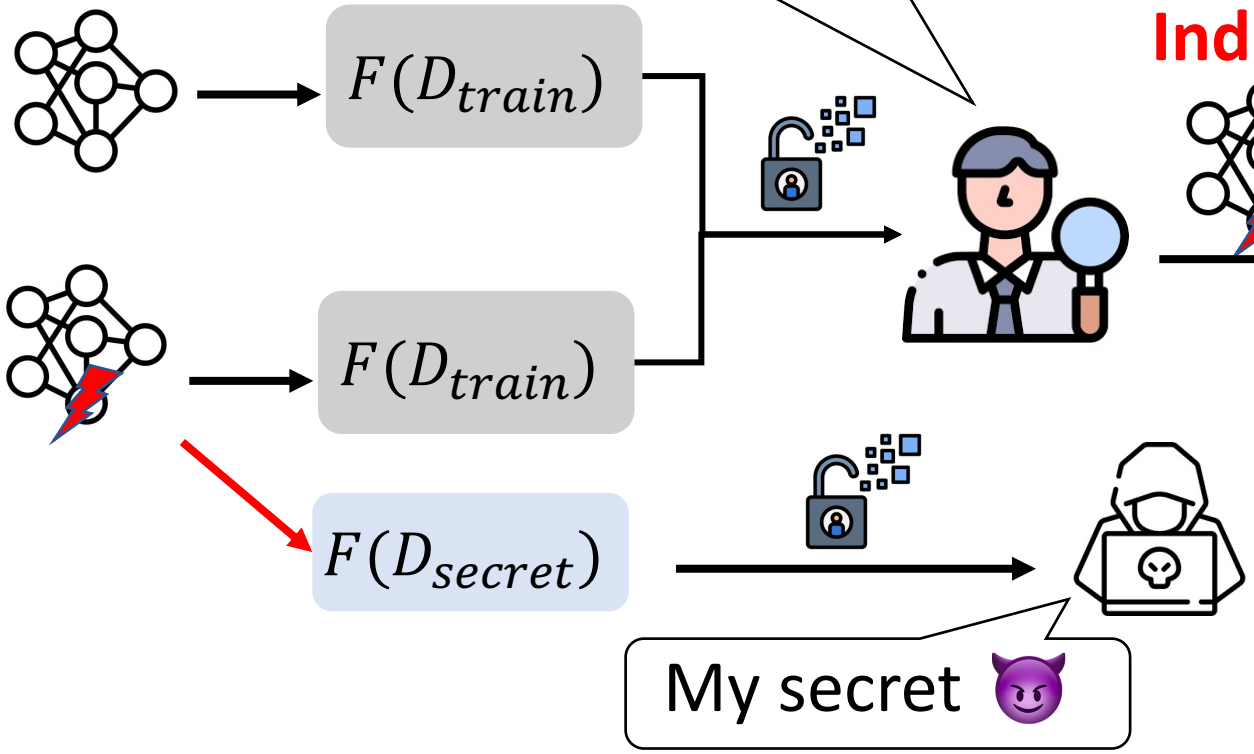
	Best prior active attack	Our active attack
How many data are leaked?	13% → 35%	13% → 99.99%
Accuracy drop	>45%	<1%

Tramer et al., CCS'22

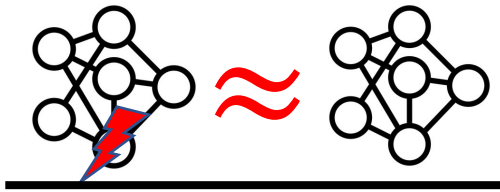
Our attack exposes the worst-case privacy leakage has minimal performance impact can disguise high privacy leakage

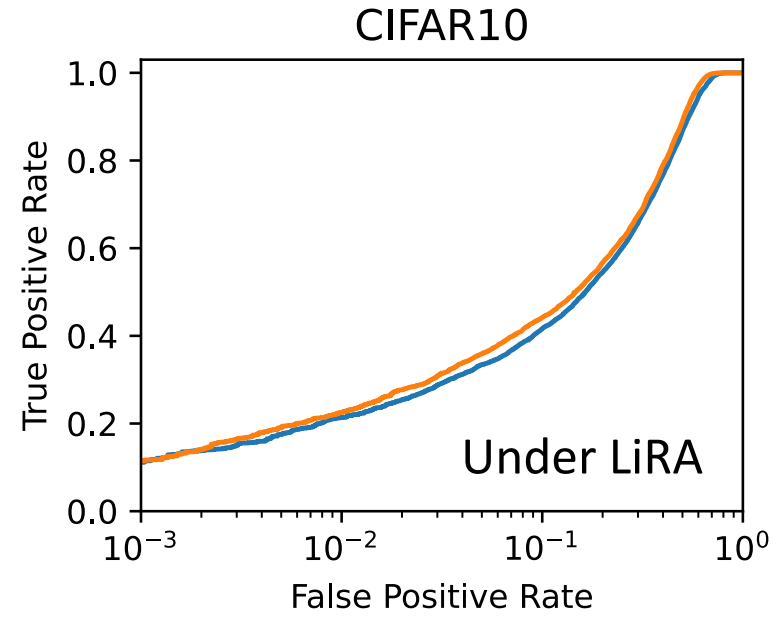
Artifact Evaluated
NDSS SYMPOSIUM
Available
Functional
Reproduced

Use existing tools to audit privacy leakage



Indistinguishable!





Carlini et al., S&P'22

Conclusion

Zitao Chen
zitaoc@ece.ubc.ca

Using third-party **ML codebase** has hidden privacy risk

New direction to construct **stealthy attacks** and inflict **worst-case leakage**

The **first** result → **Existing privacy auditing** methods can be **unreliable**!