PANORAMIA: Privacy Auditing of Machine Learning Models without Retraining

Mishaal Kazmi, Hadrien Lautraite, Alireza Akbari, Qiaoyue Tang, Mauricio Soroco,
Tao Wang, Sébastien Gambs, Mathias Lécuyer

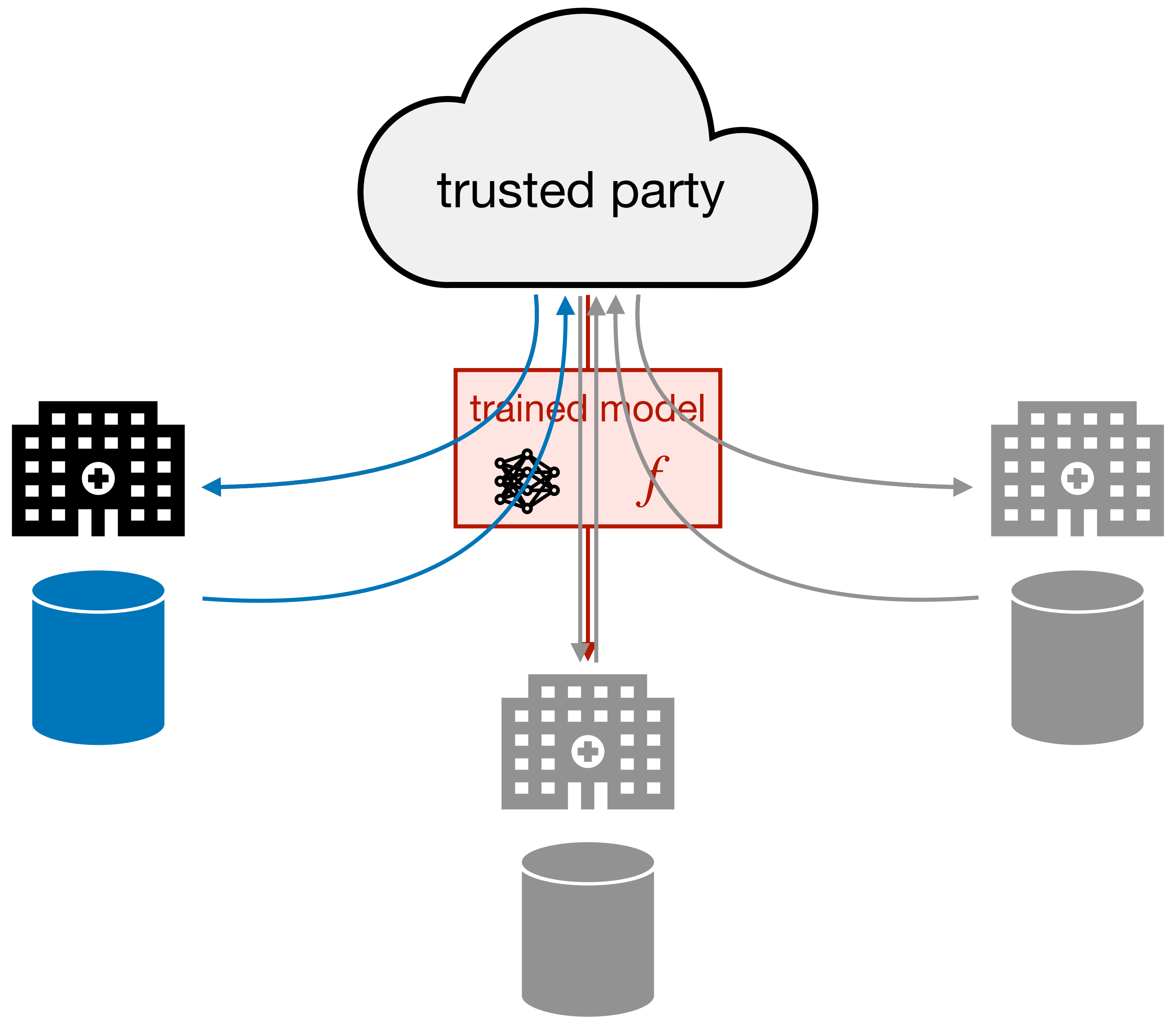


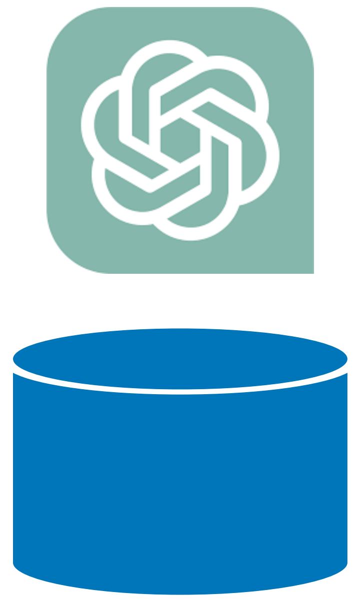



Challenging privacy auditing settings

Consider a data contributor (e.g., hospital, bank, consumer) co-training a model with other participants.

Or a foundation model, trained on all the data in the world.





How do we assess privacy risks of these models, on their training data?

Differential Privacy (DP) quantifies Privacy Loss

Hypothesis test definition of DP [Dong et al. 2019]:

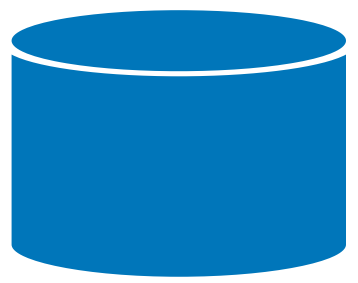
> We can frame privacy as a hypothesis test between $\mathscr{H}_0 : x \in D$ and $\mathscr{H}_1 : x \notin D$ (i.e. whether x is in training data D).

> This hypothesis test is a membership inference attack (MIA).

> DP implies a bound on the power of such hypothesis tests: any test based on an ε-DP has TPR $\leq e^{\varepsilon}$FPR.

Privacy measurement with MIAs

What does it mean for a privacy auditor?



For each datapoint x:

- Train f with or without x;
- Run a MIA to guess if x was in the training set or not.

If TPR $\leq e^{\varepsilon}$FPR, then f is not consistent with an ε-DP algorithm.

Privacy measurement with MIAs

What does it mean for a privacy auditor?

Practical issues:

> Needs to retrain model f;

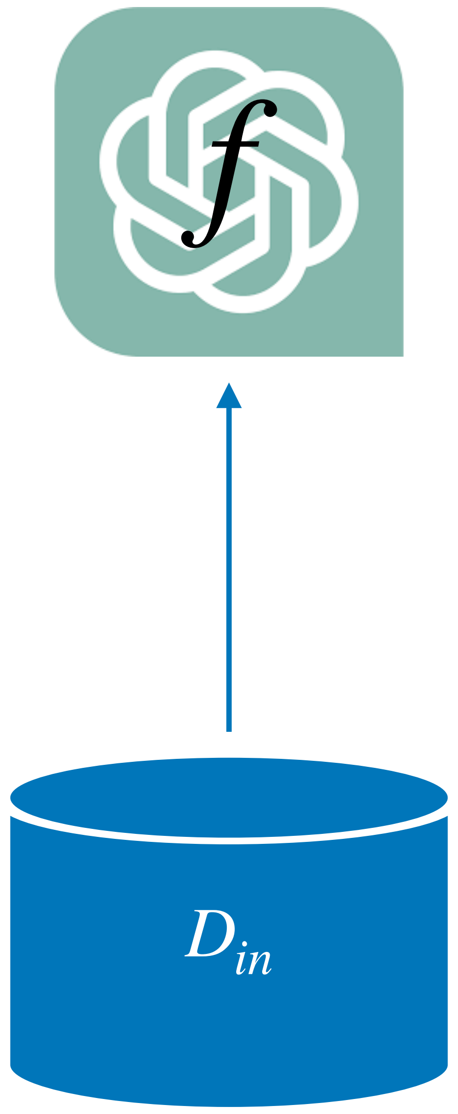
> Needs datapoints removed from the training set, so we're changing the model;

> It's typical to "poison" the model to make the algorithm audit more efficient: not what we want here!

PANORAMIA:
Privacy Audits without Model Retraining

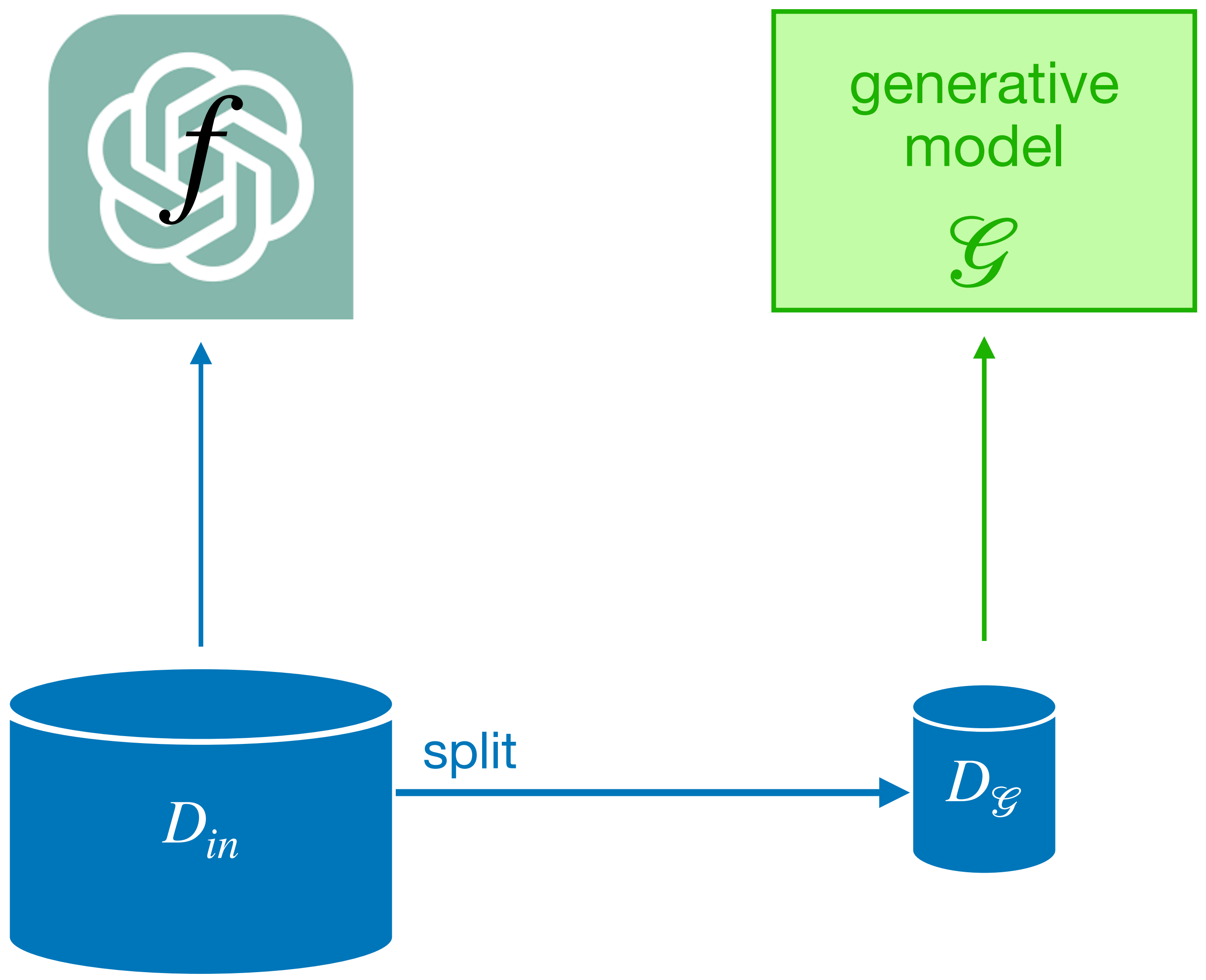
PANORAMIA overview

Remember that we want to **audit model** f for a specific subset of the training **data** D_{in}.



PANORAMIA overview

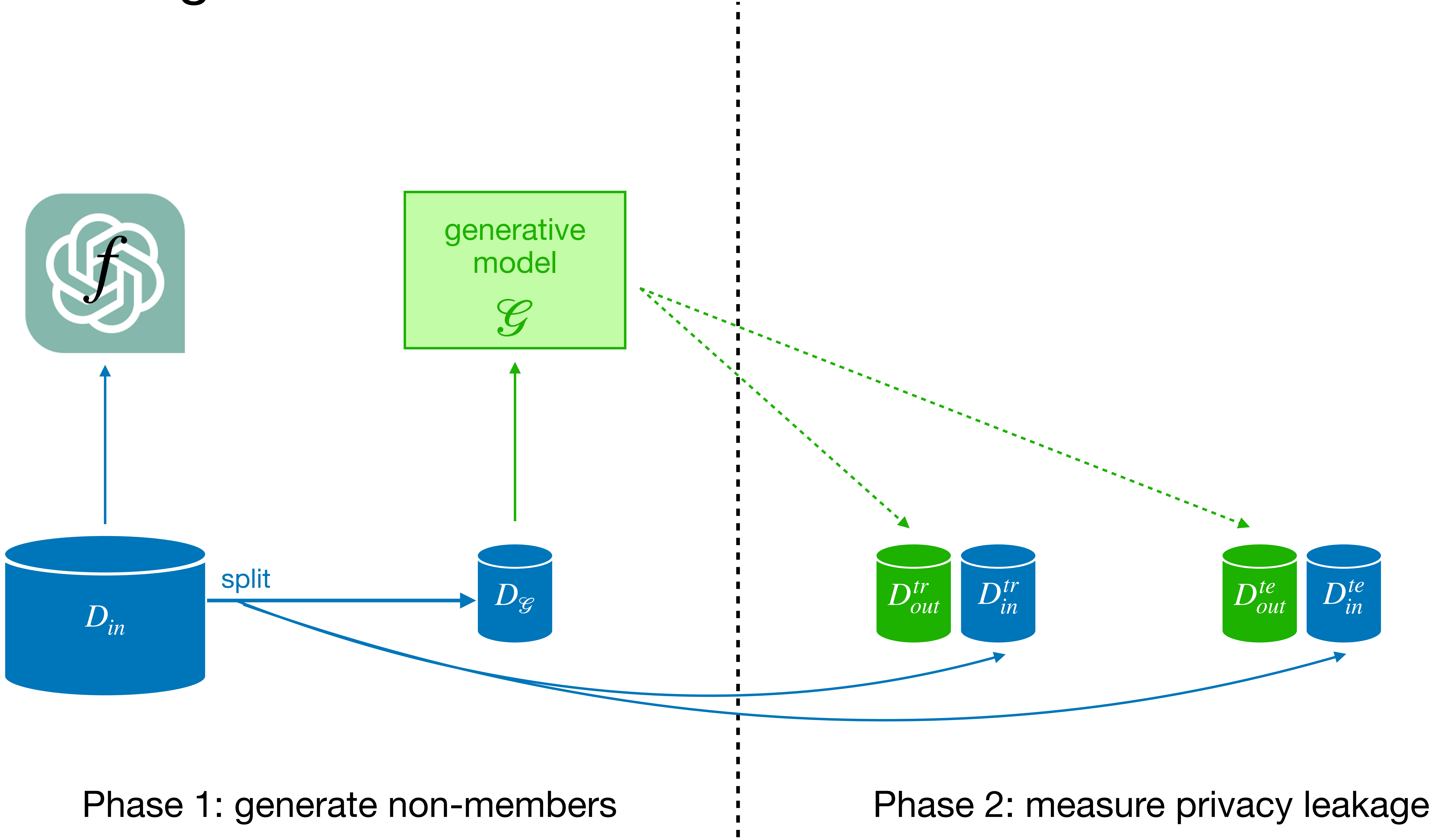
We train to generate non-member data using a subset of D_{in}.



Phase 1: generate non-members

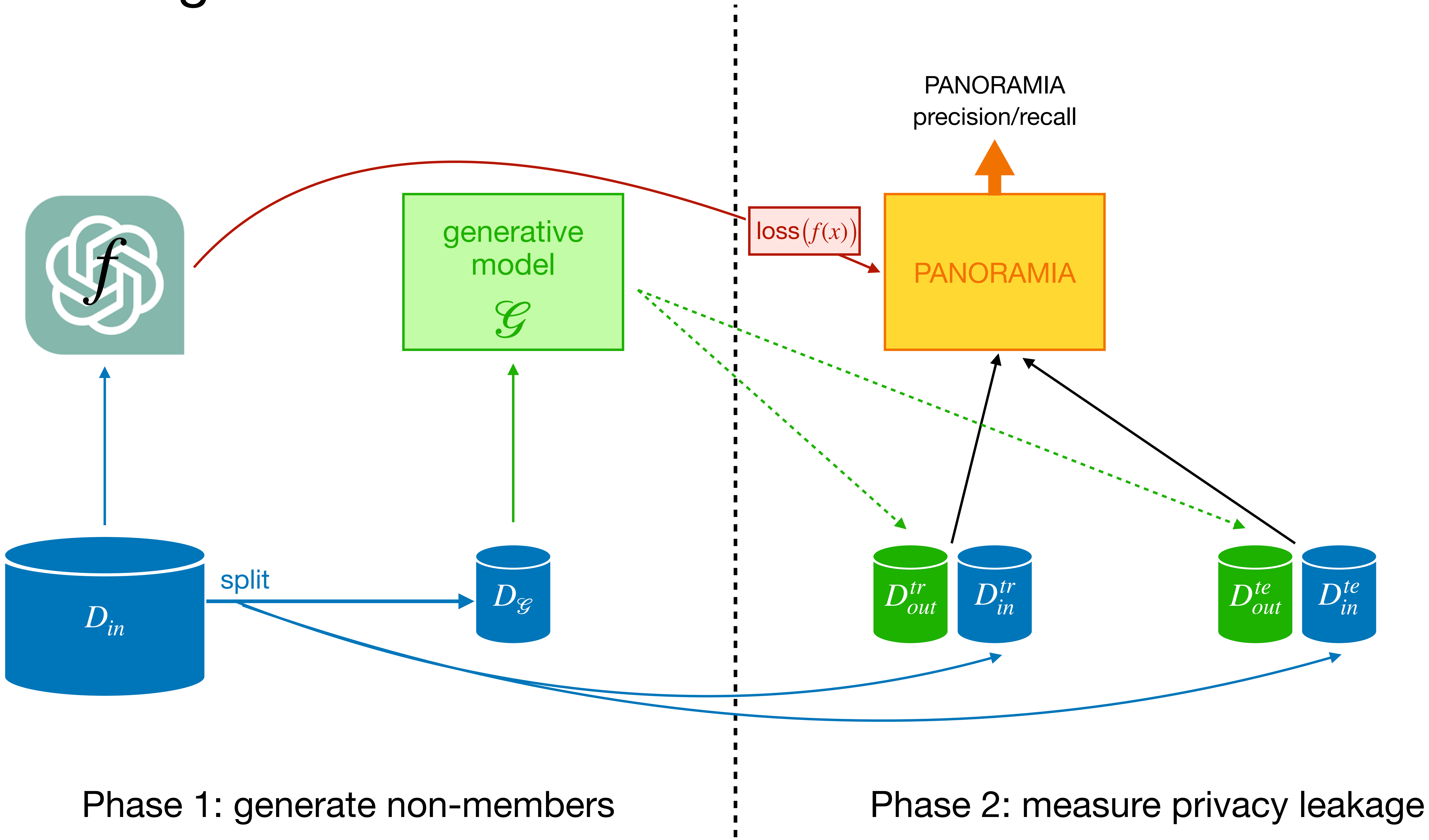
PANORAMIA overview

Using generated non-members, we train a Membership Inference Attack and evaluate it on a large test set.



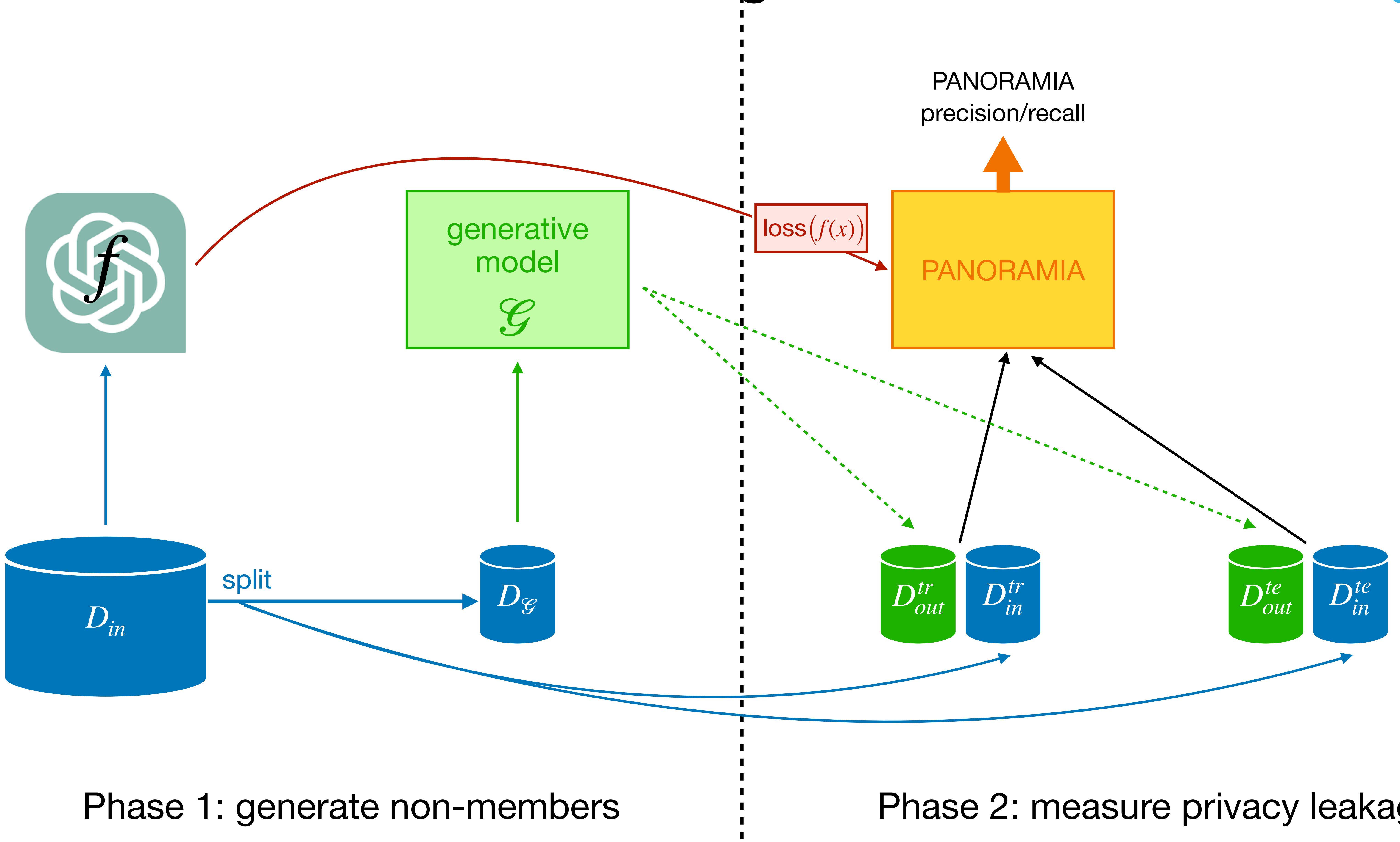
PANORAMIA overview

Using generated non-members, we train a Membership Inference Attack and evaluate it on a large test set.



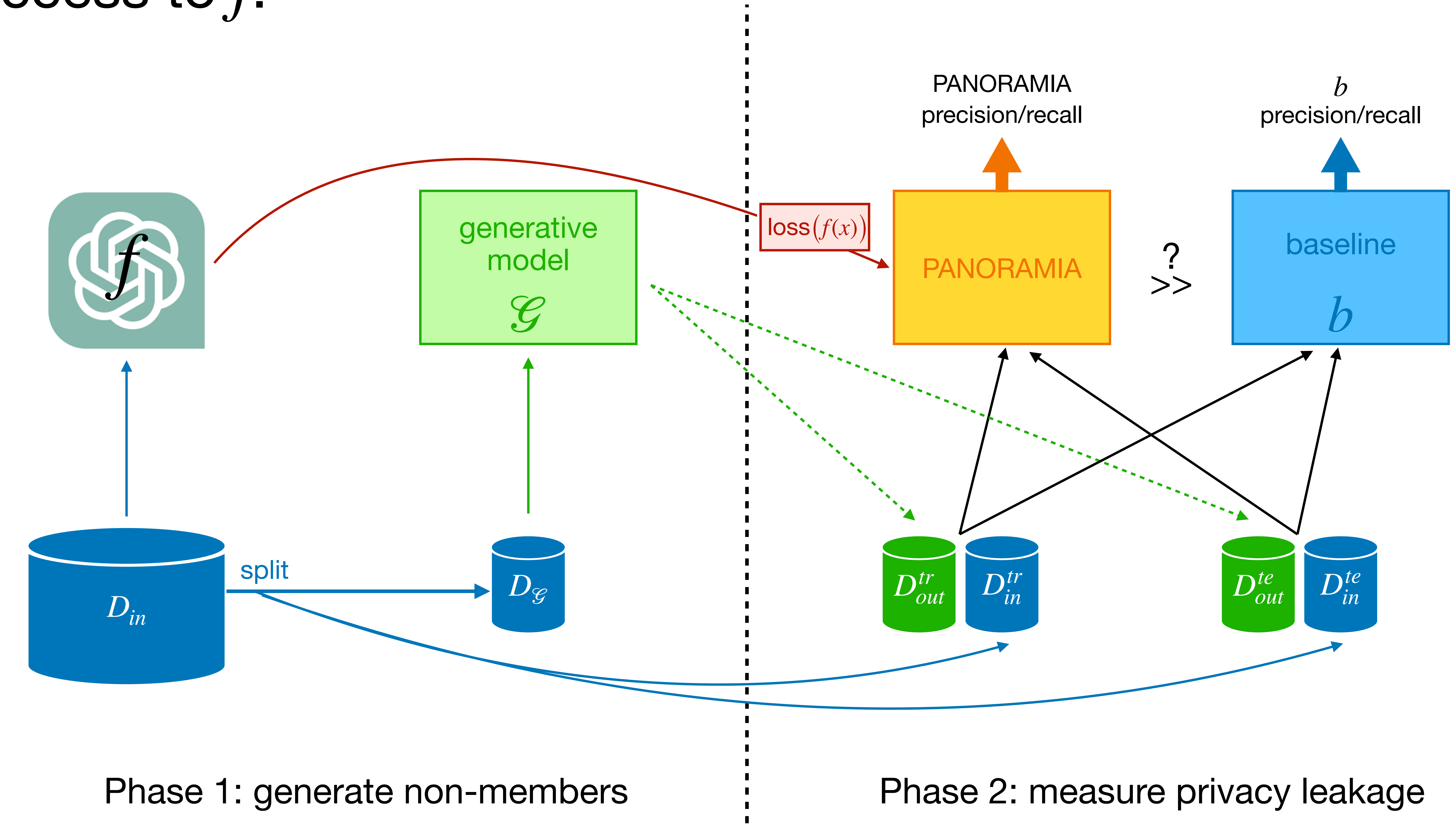
PANORAMIA overview

Using generated non-member and member data, we train a Membership Inference Attack and evaluate it on a large test set. What is wrong here?



PANORAMIA overview

We need to compare our MIA results to that of a baseline model (b) that does not have access to f.



Quantifying privacy leakage with PANORAMIA

We adapt O(1) “averaging over data” results (Steinke et al. 2023) to define **PANORAMIA auditing game**:

$$s \sim \text{Bernoulli}\left(\frac{1}{2}\right)^m, s_i \in \{0,1\},$$

$$x_i = (1 - s_i)x_i^{gen} + s_i x_i^{in}, \forall i \in \{1,\ldots,m\}, \text{ put } x_i \in D,$$

$$\text{Predict membership } T_i \in \mathbb{R}^+, \forall i.$$

Source: Thomas Steinke, Milad Nasr, and Matthew Jagielski. ‘Privacy Auditing with One (1) Training Run’. In: NeurIPS, 2023

Quantifying privacy leakage with PANORAMIA

We measure the **generator quality (c)** using the baseline model b:

For all $c > 0$, we say that a generative model $\mathcal{G}$ is c-close for data distribution $\mathcal{D}$ if:

$$\forall x \in \mathcal{X},\ e^{-c}\mathbb{P}_{\mathcal{D}}[x] \leq \mathbb{P}_{\mathcal{G}}[x]$$

Quantifying privacy leakage with PANORAMIA

The baseline gives us a test for c which we can get a **lower-bound** c_{lb}:

$$\mathbb{P}_{S,X,T^b}\left[\sum_{i=1}^{m} T_i^b \cdot S_i \geq v \mid T^b = t^b\right] \leq \mathbb{P}_{S' \sim \text{Bernoulli}\left(\frac{e^c}{1+e^c}\right)^m}\left[\sum_{i=1}^{m} t_i^b \cdot S_i' \geq v\right]$$

Quantifying privacy leakage with PANORAMIA

MIA gives us a test for leakage through both f and the difference between $\mathscr{D}$ and $\mathscr{G}$ which we can get a **lower-bound** $\{c + \varepsilon\}_{lb}$:

$$\mathbb{P}_{S,X,T^a}\left[\sum_{i=1}^{m} T_i^a \cdot S_i \geq v \mid T^a = t^a\right] \leq \mathbb{P}_{S' \sim \text{Bernoulli}\left(\frac{e^{c+\epsilon}}{1+e^{c+\epsilon}}\right)^m}\left[\sum_{i=1}^{m} t_i^a \cdot S_i' \geq v\right]$$

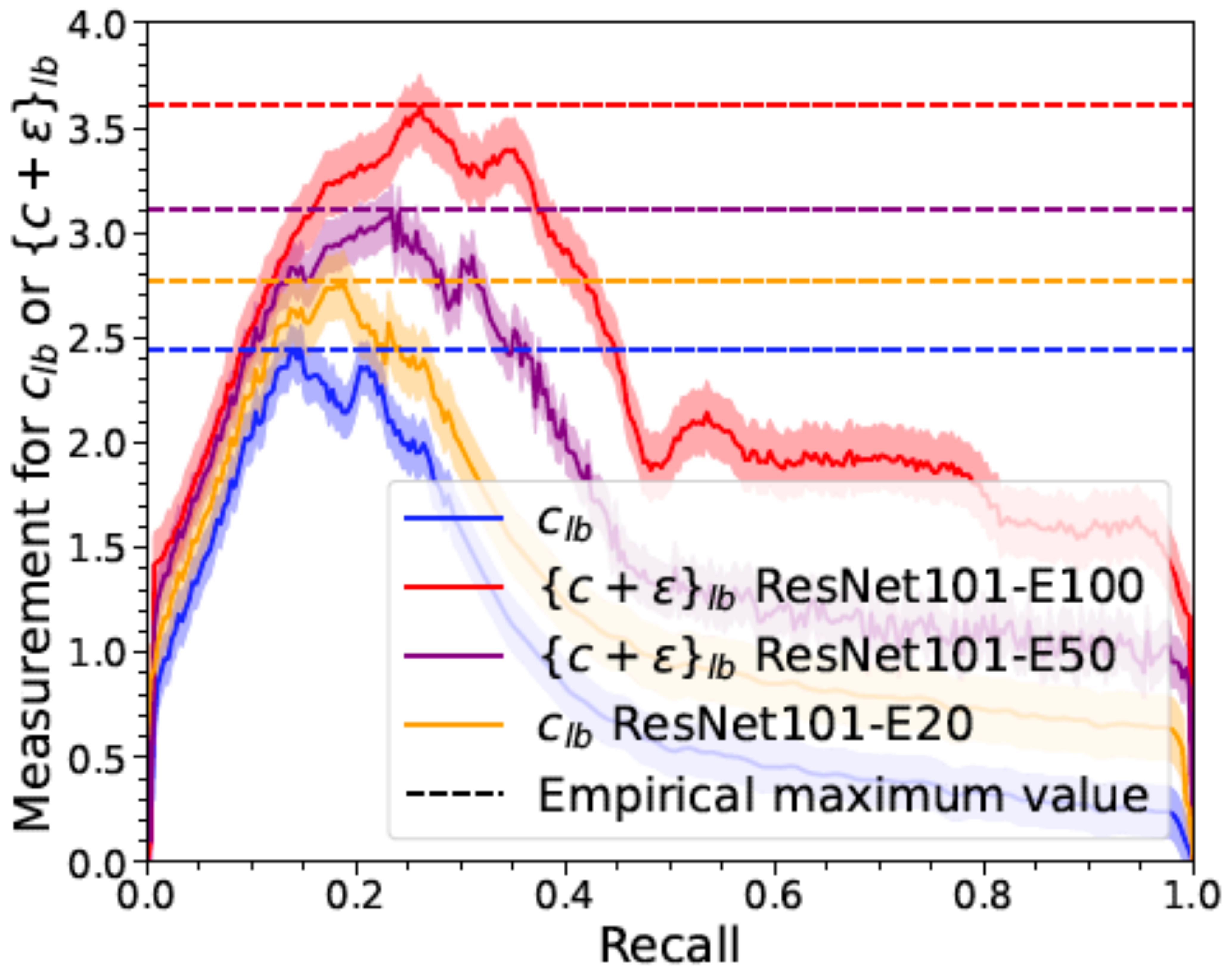
Quantifying privacy leakage with PANORAMIA

We use $\tilde{\varepsilon} = \{c + \varepsilon\}_{lb} - c_{lb}$ as an estimate of privacy leakage.

"The generator $\mathcal{G}$ could be c-good, and if it is, then f is no better than ε-DP as far as its leakage of D_{in} is concerned."

Empirical results: ResNet101 on CIFAR-10

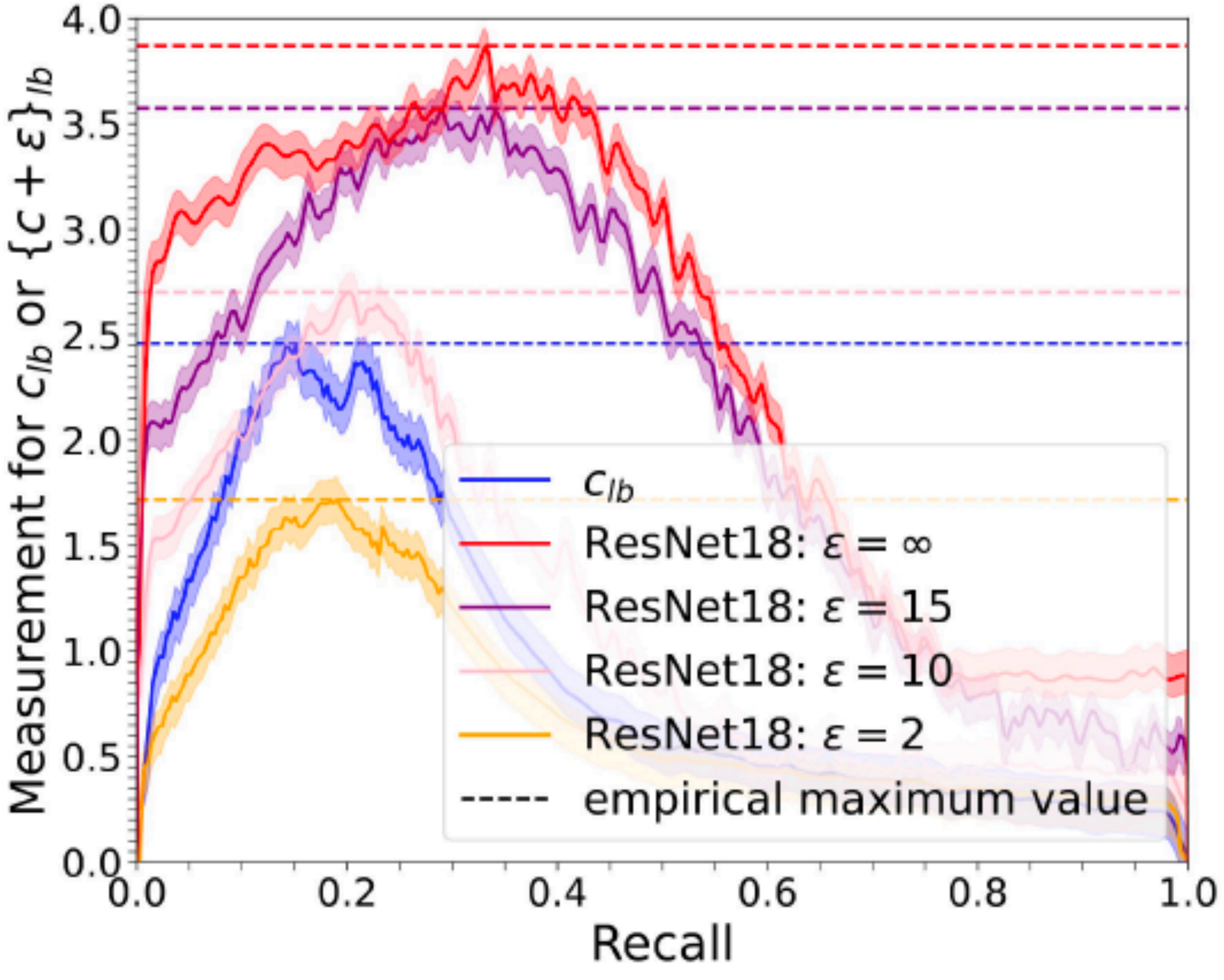
(c) CIFAR10 Real Images (d) CIFAR10 Synthetic Images



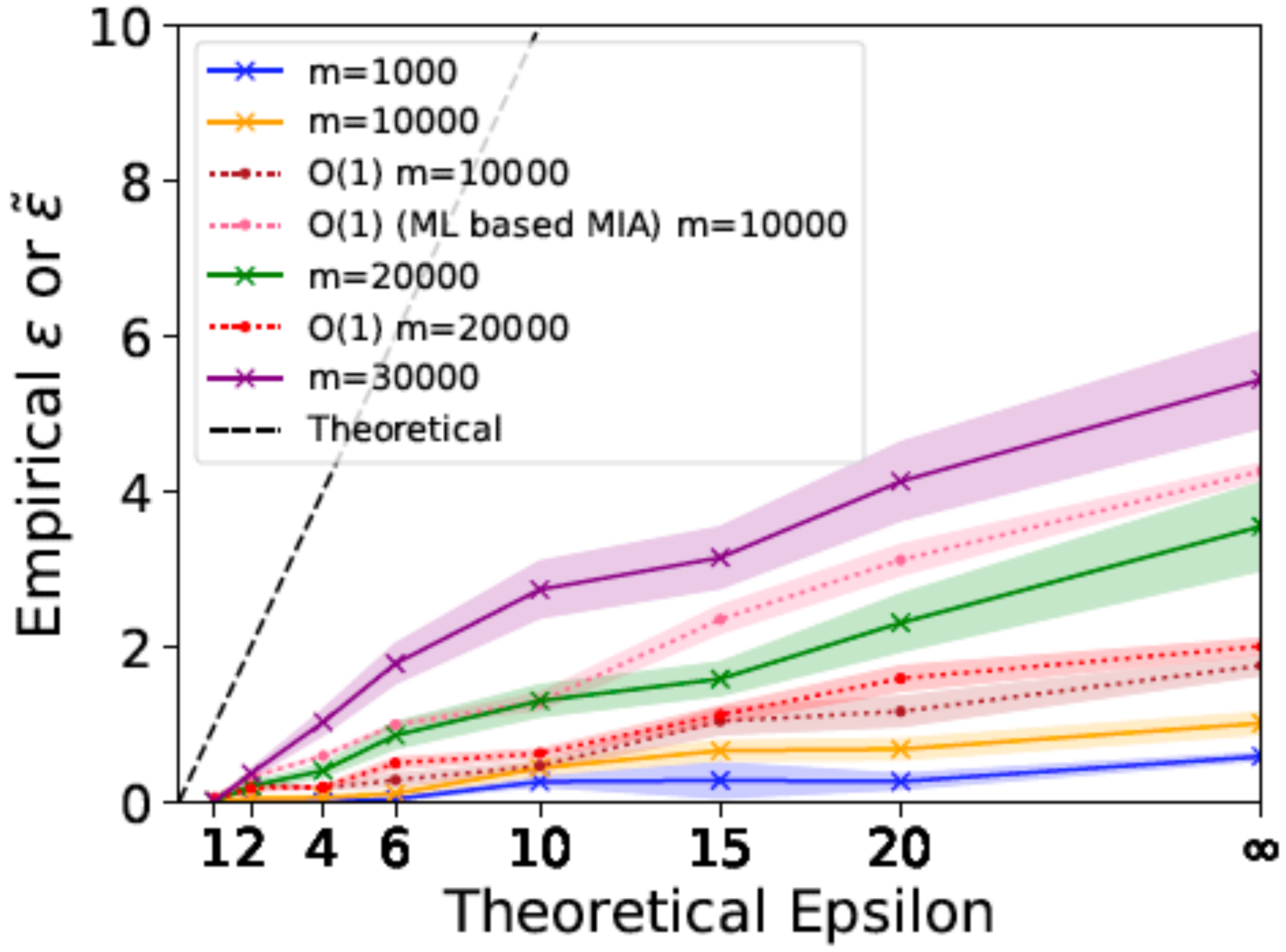
Baseline works well

Able to detect meaningful amounts of privacy loss

Empirical results: DP models on CIFAR-10



Able to detect larger privacy loss with DP models of larger ϵ



Able to use more data to increase the amount of leakage we can measure

Conclusion

> We can audit ML models and specific subsets of their training set with no control over the training pipeline.

> Empirically, results are close to those of state-of-the art methods (that do require changing the training data and/or retraining the model).

> Full paper: https://arxiv.org/abs/2402.09477

> Code repository: https://github.com/ubc-systopia/panoramia-privacy-measurement