GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models

Dingfan Chen[1], Ning Yu[2,3], Yang Zhang[1], Mario Fritz[1]

[1]CISPA Helmholtz Center for Information Security, Germany
[2]Max Planck Institute for Informatics, Germany
[3]University of Maryland, College Park

Motivation

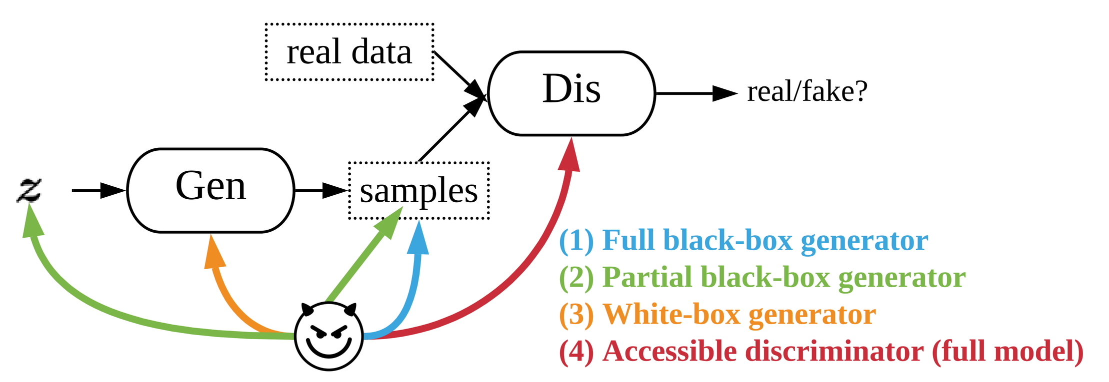
- Generative adversarial Networks (GANs) have been largely used on privacy sensitive datasets, e.g., face images and medical records
- However, existing works mainly focus on attacks against discriminative models and the privacy risk of generative models have not yet been investigated systematically
- **Our work:** Membership Inference Attack against GANs (whether a query sample has been used to train a GAN model?)
- Crucial to understand and control privacy leakage; provides insights for privacy-preserving data sharing

Contributions

- **Taxonomy**
  - Categorize attack scenarios against generative models
  - Benchmark future research
- **Novel attack models**
  - Generic; easy-to-implement; effective; theoretically grounded
- **Extensive evaluation**
  - 3 datasets with diverse data modalities, 5 victim models, 4 attack scenarios …

Taxonomy

- What information does the attacker know?
  - White-box □/black-box ■?
  - Which GANs' components are accessible?
  (z: latent code; **Gen**: Generator; **Dis**: Discriminator)



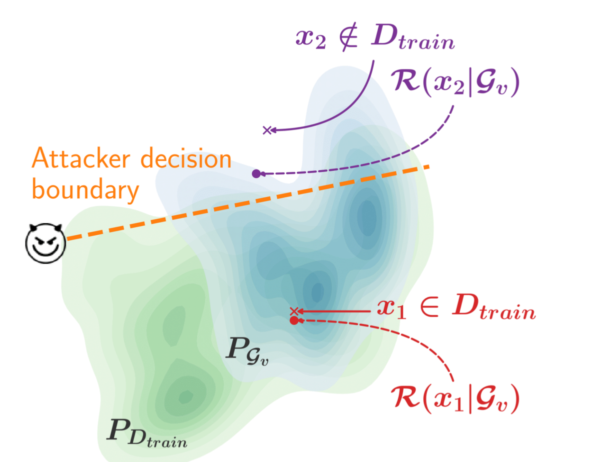
	Latent code	Generator	Discriminator
(1) Full black-box generator[1,2]	✗	■	✗
(2) Partial black-box generator	✓	■	✗
(3) White-box generator	✓	□	✗
(4) Accessible discriminator (full model)[1]	✓	□	✓

Generic Attack Model

Attacker finds the **best reconstruction** of a query sample given **different types of access** to the victim generator.

- **Insight**:
  Smaller reconstruction error for training data.
- **Generic Model:**
  Optimization problem

$$\mathcal{R}(x|\mathcal{G}_v) = \mathcal{G}_v(z^*)$$
$$z^* = \underset{z}{\operatorname{argmin}}\, L(x, \mathcal{G}_v(z))$$



- **Objective:**

$$\underset{z}{\text{minimize}} \quad L(x, \mathcal{G}_v(z)) = \lambda_1 L_2(x, \mathcal{G}_v(z)) + \lambda_2 L_{\text{lpips}}(x, \mathcal{G}_v(z)) + \lambda_3 L_{\text{reg}}(z)$$

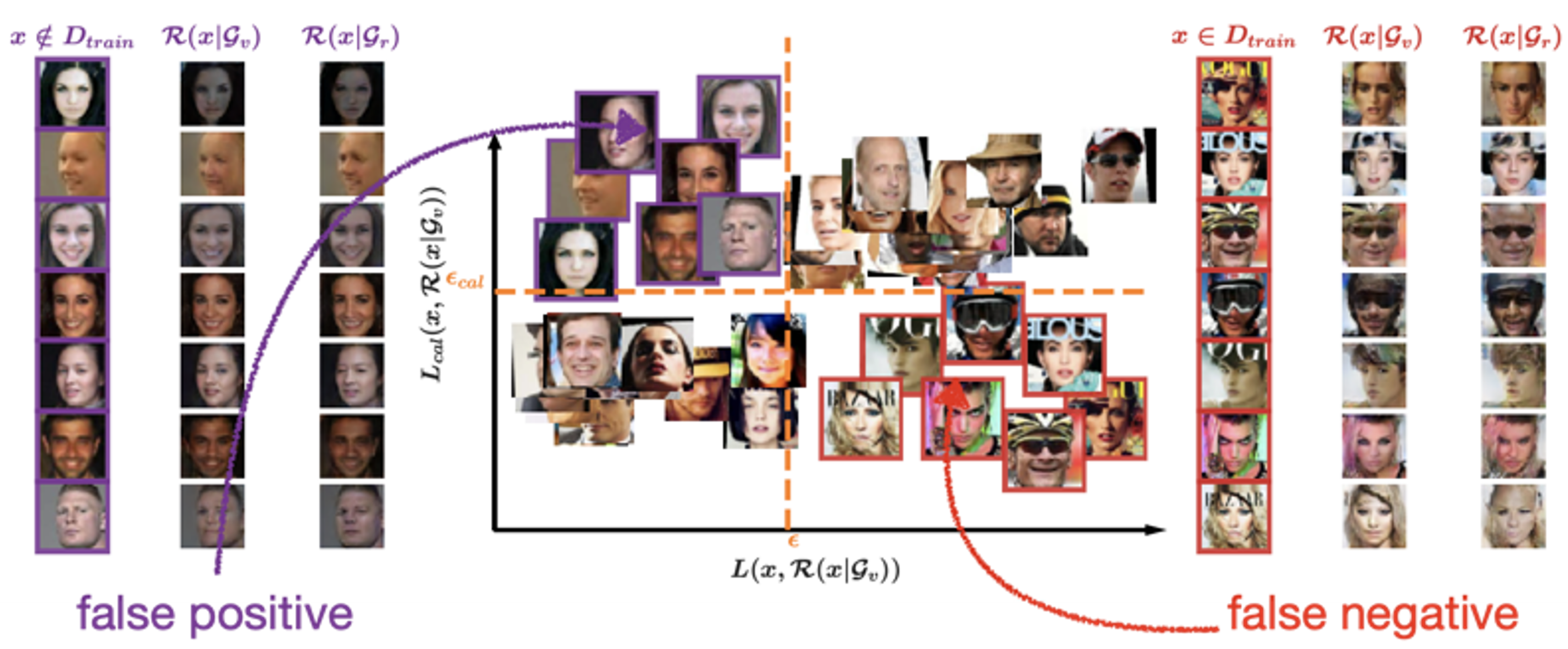
$$\text{where} \quad L_2(x, \mathcal{G}_v(z)) = \|x - \mathcal{G}_v(z)\|_2^2$$
$$L_{\text{reg}}(z) = (\|z\|_2^2 - dim(z))^2$$

- **Different types of access:**

(1) Full black-box generator	KNN search
(2) Partial black-box generator	Powell's conjugate direction method
(3) White-box generator	L-BFGS quasi-Newton method

Attack Calibration

- **Problem:**
  the reconstruction error is query-dependent ('hard' samples, underrepresented samples)

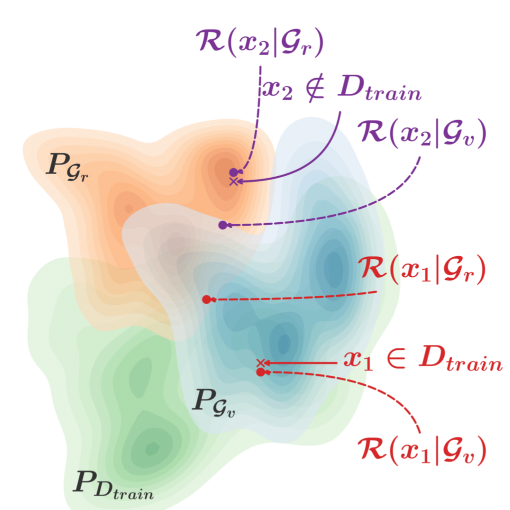


- **Solution:** Attack Calibration

$$L_{\text{cal}}(x, \mathcal{R}(x|\mathcal{G}_v)) = L(x, \mathcal{R}(x|\mathcal{G}_v)) - L(x, \mathcal{R}(x|\mathcal{G}_r))$$

victim model   reference model

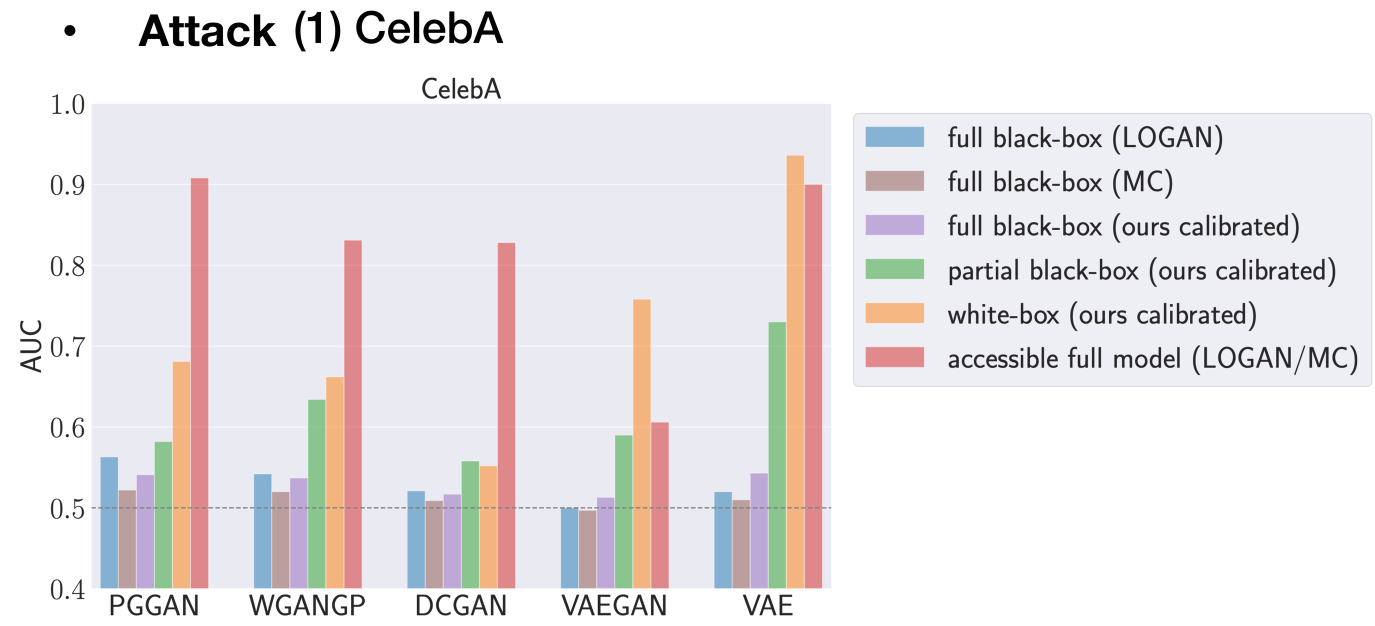
  - Train a **reference model** with:
    - **relevant** but **disjoint** dataset
    - **irrelevant** network architecture to victim model



- **Theory:** near-optimal under a Bayesian perspective[3]

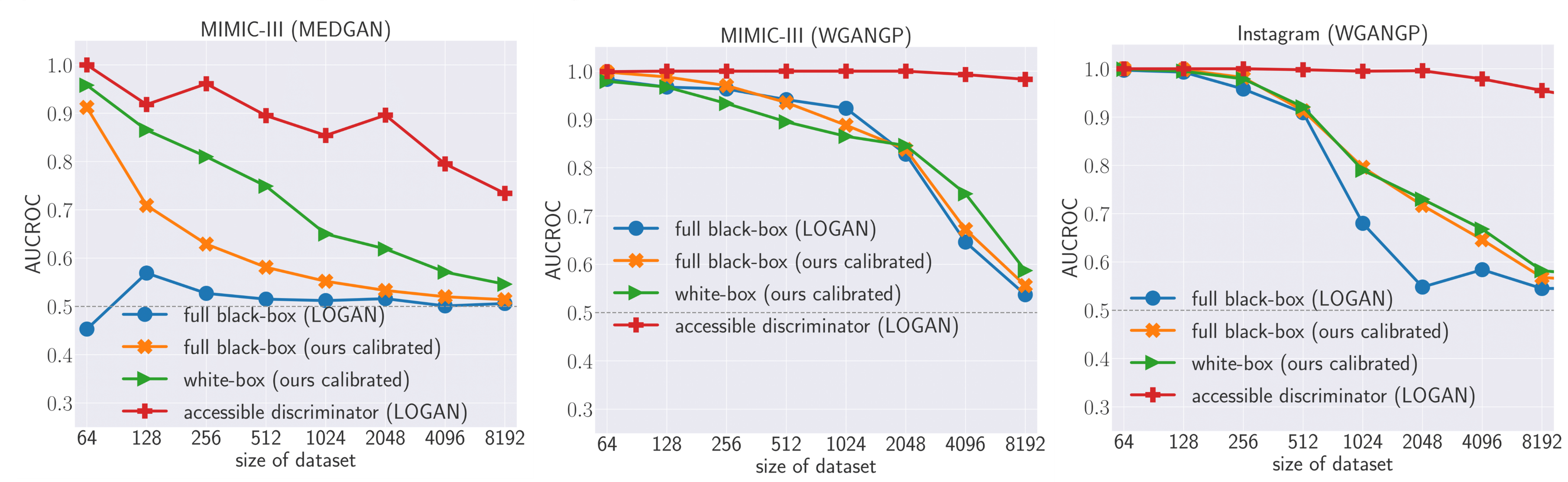
Experiment results

- **3 Datasets:**
  CelebA (face), MIMIC III (medical), Instagram (location)
- **5 GAN Models:**
  PGGAN, WGANGP, DCGAN, VAEGAN, MedGAN
- **2 Baselines:**
  LOGAN[1], MC[2]
- **Results:**
  - **Attack** (1) CelebA

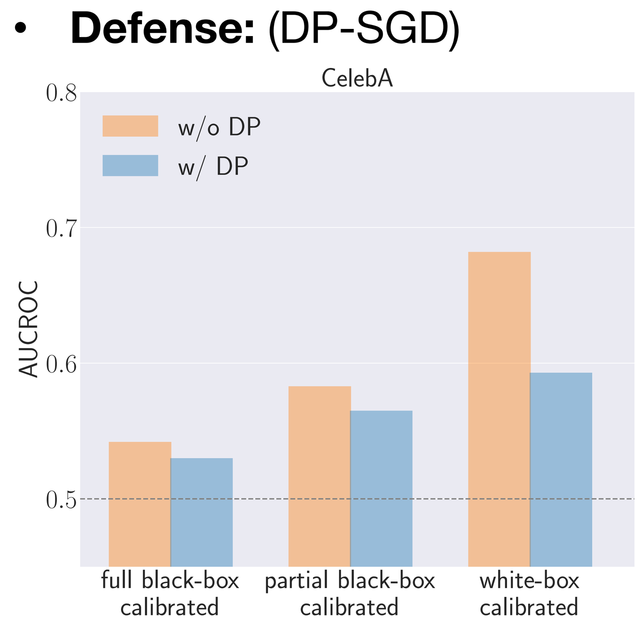


[1] Hayes et al., "LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks", PoPETs 2019
[2] Hilprecht et al., "Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models", PoPETs 2019
[3] Sablayrolles et al., "White-box vs Black-box: Bayes Optimal Strategies for Membership Inference", ICML 2019

(2) MIMIC III, Instagram (non-image dataset)



- **Defense:** (DP-SGD)



Summary

- A simple learning-free attack model works sufficiently well
- Attack performance highly depends on:
  - The size of the dataset
  - Model structure
  - Amount of knowledge about the victim model
- Differential privacy defense is effective against real-world MI attack but compromises utility and efficiency
- Code and models are available on Github:
  https://github.com/DingfanChen/GAN-Leaks