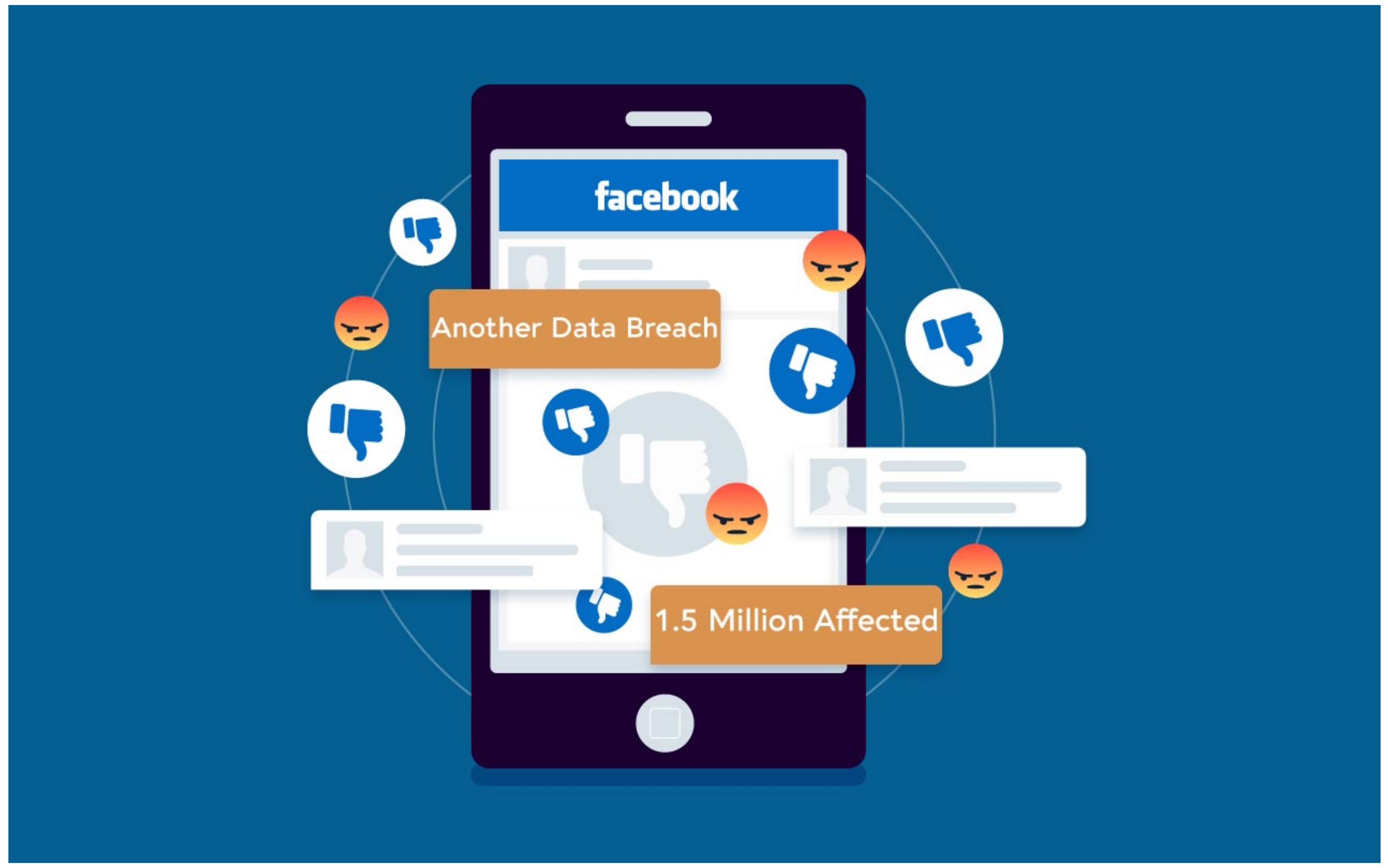
COMP6211I: Trustworthy Machine Learning

Confidentiality (defense)

Minhao CHENG

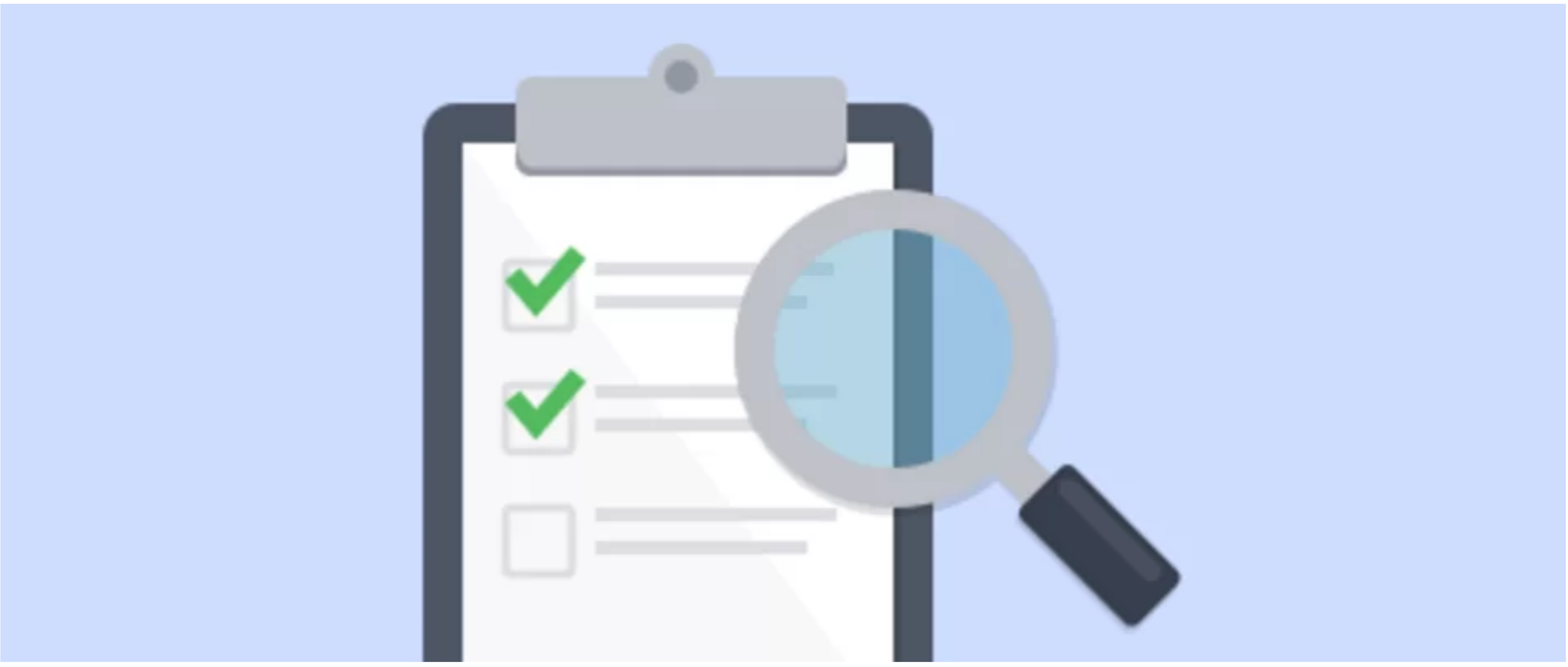
Privacy problem

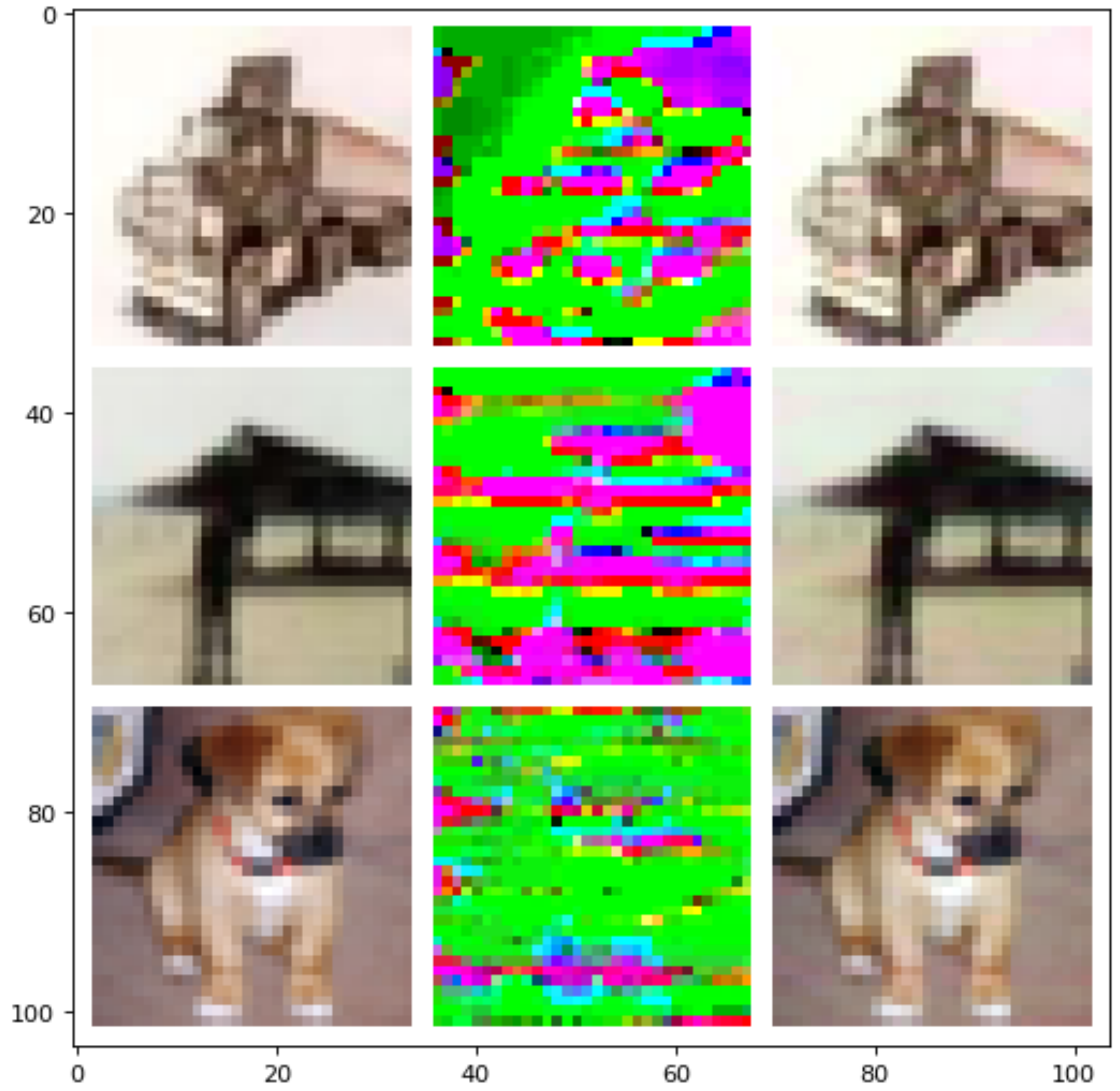
- Datasets are collected without mutual consent
- Datasets are vulnerable to steal for training other models



Privacy protection

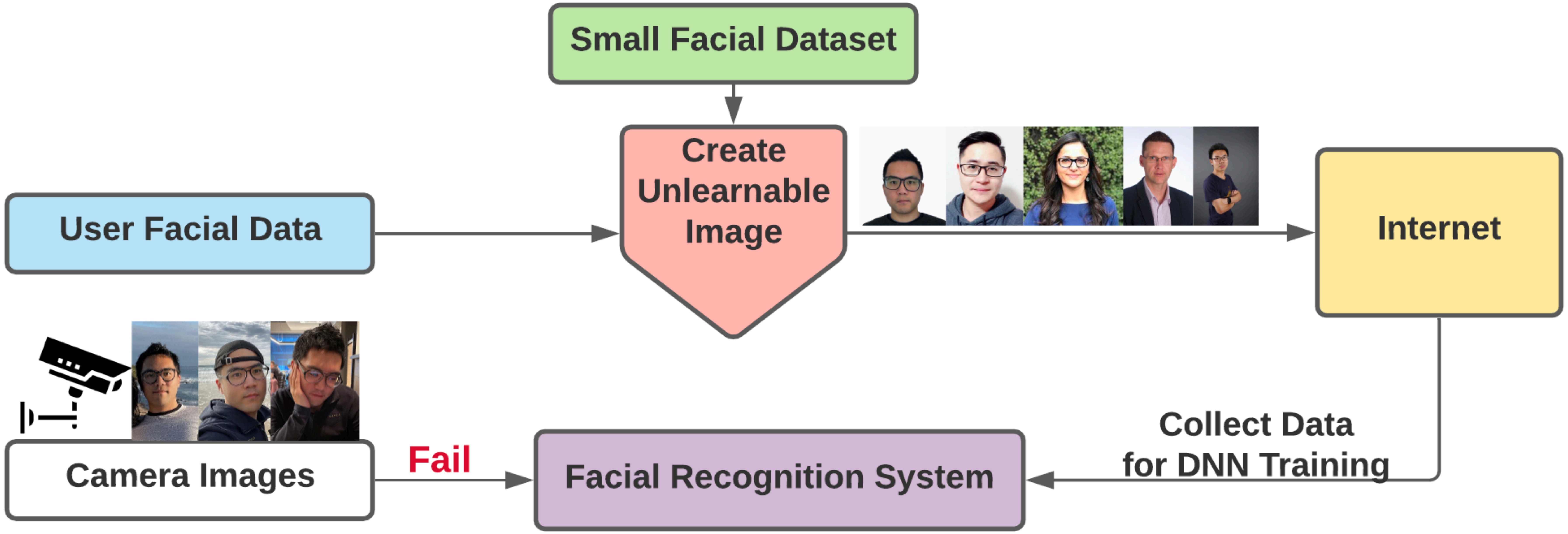
- How we could prevent other to use your personal data?
  - Persevering privacy by obfuscating information from the dataset
  - Proof their usage of your data





Unlearnable example

- Make the example unlearnable should not affect its quality for normal usage
- Noise could only be added prior to model training



Threat model

- Defender has full access to the data
- Cannot interfere with training and don’t have access to the full training dataset
- Cannot further modify data once the examples are created

Problem formulation

- Clean training datasets $\mathscr{D}_c$ and testing $\mathscr{D}_t$
- Transform training data $\mathscr{D}_c$ into unlearnable $\mathscr{D}_u$ so that DNNs trained on $\mathscr{D}_u$ will perform poorly on $\mathscr{D}_t$
- $\mathscr{D}_c = \{(\mathbf{x_i}, y_i)\}_{i=1}^n, \mathscr{D}_u = \{(\mathbf{x_i'}, y_i)\}_{i=1}^n$, where $\mathbf{x}' = \mathbf{x}' + \delta$
- $\delta \in \Delta \in \mathbb{R}^d$ should be “invisible”
  - A choice would be $||\delta||_p \leq \epsilon$

Problem formulation

Objective

- Trick the model into learning a strong correlation between and noise and the labels when trained on $\mathscr{D}_u$:
  - arg $\min_{\theta} \mathbb{E}_{(\mathbf{x}',y)\sim\mathscr{D}_u} L(f(\mathbf{x}', y))$
  - Noise: $\mathbf{x}'_i = \mathbf{x}'_i + \delta_i,$
    - Sample-wise: $\delta_i \in \Delta_s = \{\delta_1, \ldots, \delta_n\}$
    - Class-wise: $\delta_y i \in \Delta_c = \{\delta_1, \ldots, \delta_K\}$

Problem formulation

Objective

- A simplified way
  - $\arg\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}_c}[\min_{\delta} L(f'(\mathbf{x}' + \delta, y)]$ s.t. $\|\delta\|_p \leq \epsilon$
    - Where f' denotes the source model used for noise generation

Problem formulation

Objective

- A simplified way
  - $\arg\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_c}[\min_{\delta} L(f'(\mathbf{x}'+\delta, y)]$ s.t. $\|\delta\|_p \leq \epsilon$
    - Where f' denotes the source model used for noise generation
- Sample-wise: use PGD

$$\boldsymbol{x}'_{t+1} = \Pi_\epsilon\big(\boldsymbol{x}'_t - \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}}\mathcal{L}(f'(\boldsymbol{x}'_t), y))\big)$$

- Class-wise: use UAP on the class by accumulates the perturbation

Comparison

Sample-wise vs class-wise

- Work in different way:
  - Sample-wise:
    - Low-error samples can be ignored
  - Class-wise:
    - Make data not i.id.d

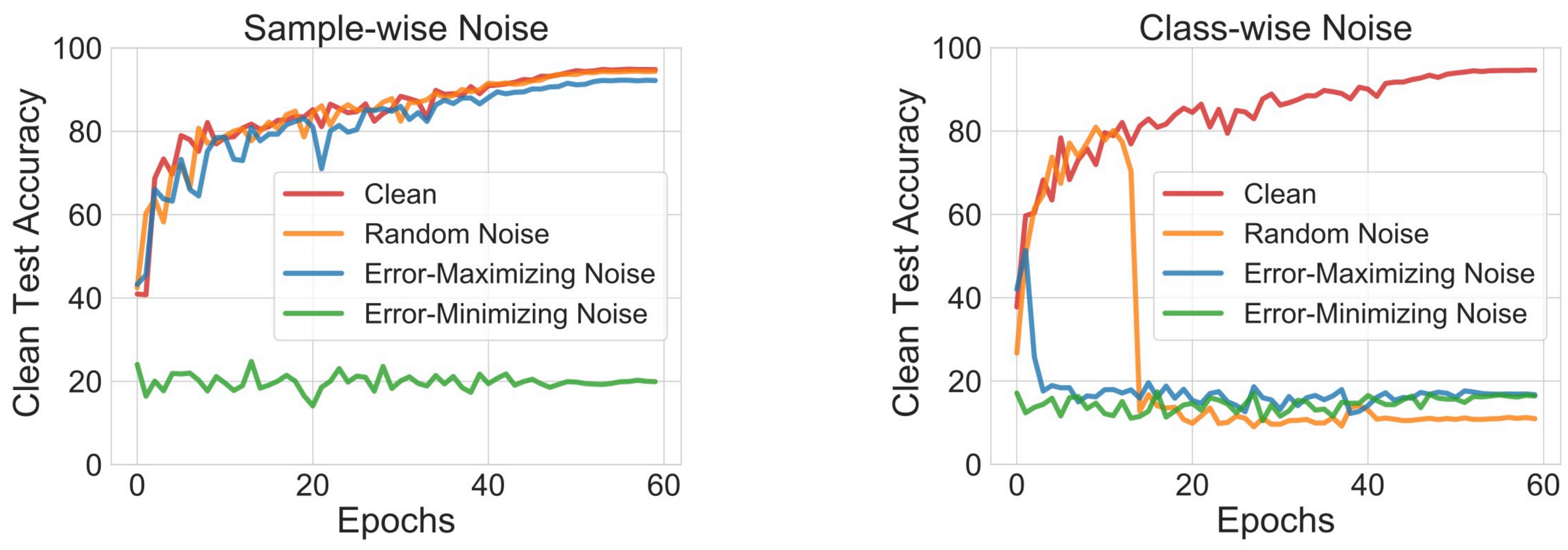


Figure 1: The unlearnable effectiveness of different types of noise: random, adversarial (error-maximizing) and our proposed error-minimizing noise on CIFAR-10 dataset. The lower the clean test accuracy the more effective of the noise.

Main results

Table 1: The top-1 clean test accuracies (%) of DNNs trained on the clean training sets ($\mathcal{D}_c$) or their unlearnable ones ($\mathcal{D}_u$) made by sample-wise (Δ_s) or class-wise (Δ_c) error-minimizing noise.

Noise Form	Model	SVHN		CIFAR-10		CIFAR-100		ImageNet*	
		$\mathcal{D}_c$	$\mathcal{D}_u$	$\mathcal{D}_c$	$\mathcal{D}_u$	$\mathcal{D}_c$	$\mathcal{D}_u$	$\mathcal{D}_c$	$\mathcal{D}_u$
Δ_s	VGG-11	95.38	**35.91**	91.27	**29.00**	67.67	**17.71**	48.66	**11.38**
	RN-18	96.02	**8.22**	94.77	**19.93**	70.96	**14.81**	60.42	**12.20**
	RN-50	95.97	**7.66**	94.42	**18.89**	71.32	**12.19**	61.58	**11.12**
	DN-121	96.37	**10.25**	95.04	**20.25**	74.15	**13.71**	63.76	**15.44**
Δ_c	VGG-11	95.29	**23.44**	91.57	**16.93**	67.89	**7.13**	71.38	**2.30**
	RN-18	95.98	**9.05**	94.95	**16.42**	70.50	**3.95**	76.52	**2.70**
	RN-50	96.25	**8.94**	94.37	**13.45**	70.48	**3.80**	79.68	**2.70**
	DN-121	96.36	**9.10**	95.12	**14.71**	74.51	**4.75**	80.52	**3.28**

* ImageNet subset of the first 100 classes.

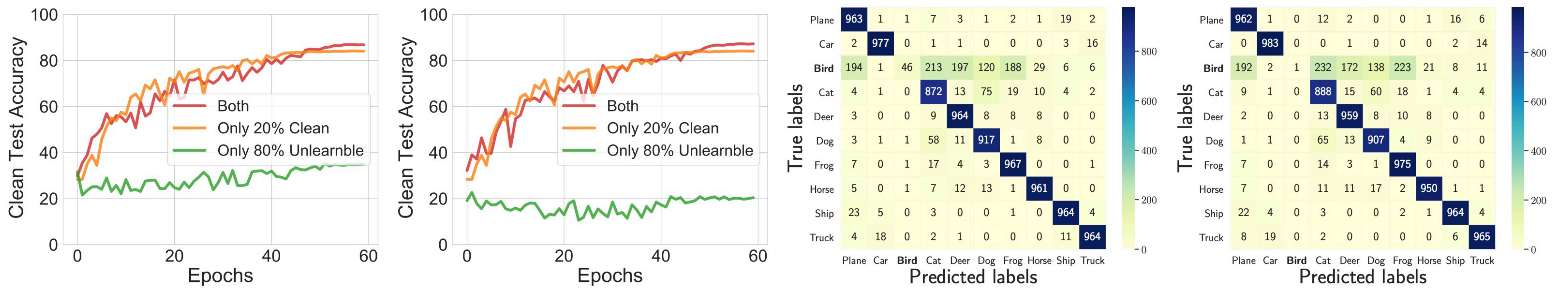
Stability

- Fail when unlearnable rate not 100%

Table 2: Effectiveness under different unlearnable percentages on CIFAR-10 with RN-18 model: lower clean accuracy indicates better effectiveness. $\mathcal{D}_u + \mathcal{D}_c$: a mix of unlearnable and clean data; $\mathcal{D}_c$: only the clean proportion of data. Percentage of unlearnable examples: $\frac{\mathcal{D}_u}{\mathcal{D}_c+\mathcal{D}_u}$.

Noise Type	**Percentage of unlearnable examples**									
	0%	**20%**		**40%**		**60%**		**80%**		**100%**
		$\mathcal{D}_u + \mathcal{D}_c$	$\mathcal{D}_c$	$\mathcal{D}_u + \mathcal{D}_c$	$\mathcal{D}_c$	$\mathcal{D}_u + \mathcal{D}_c$	$\mathcal{D}_c$	$\mathcal{D}_u + \mathcal{D}_c$	$\mathcal{D}_c$	
Δ_s	94.95	94.38	93.75	93.10	92.56	91.90	89.77	86.85	84.30	19.93
Δ_c	94.95	94.24	93.75	92.99	92.56	91.10	89.77	87.23	84.30	16.42

Single unlearnable class



(a) Sample-wise Δ_s (b) Class-wise Δ_c (c) Sample-wise Δ_s (d) Class-wise Δ_c

Figure 2: (a-b): For both sample-wise (a) and class-wise (b) noise, learning curves of RN-18 on CIFAR-10 dataset with different types of training data: 1) only 20% clean data, 2) only 80% unlearnable data, and 3) both clean and unlearnable data. (c-d): Prediction confusion matrices (on the clean test set) of two RN-18s trained on CIFAR-10 with the 'Bird' unlearnable class created by sample-wise (c) or class-wise (d) error-minimizing noise.

Against model stealing

- Watermarking model
  - Detect theft by verifying the suspect model responds with the expected outputs on watermarked inputs
  - Cons: need retraining/ vulnerable to adaptive attack
- Dataset inference: tracing the usage of your data or dataset and verification.
  - Detect the knowledge contained in the private training set of the victim

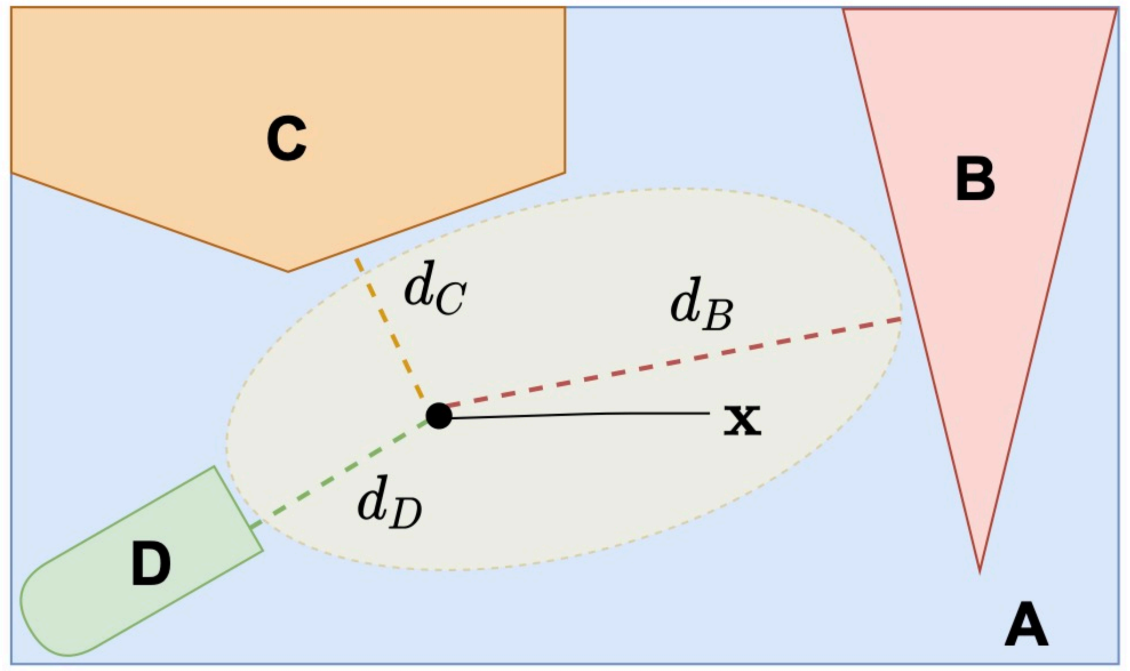
Against model stealing

- A victim $\mathscr{V}$ trains a model $f_{\mathscr{V}}$ on their private data $S_{\mathscr{V}} \subseteq \mathscr{K}_{\mathscr{V}}$ ， $\mathscr{K}_{\mathscr{V}}$ is the private knowledge

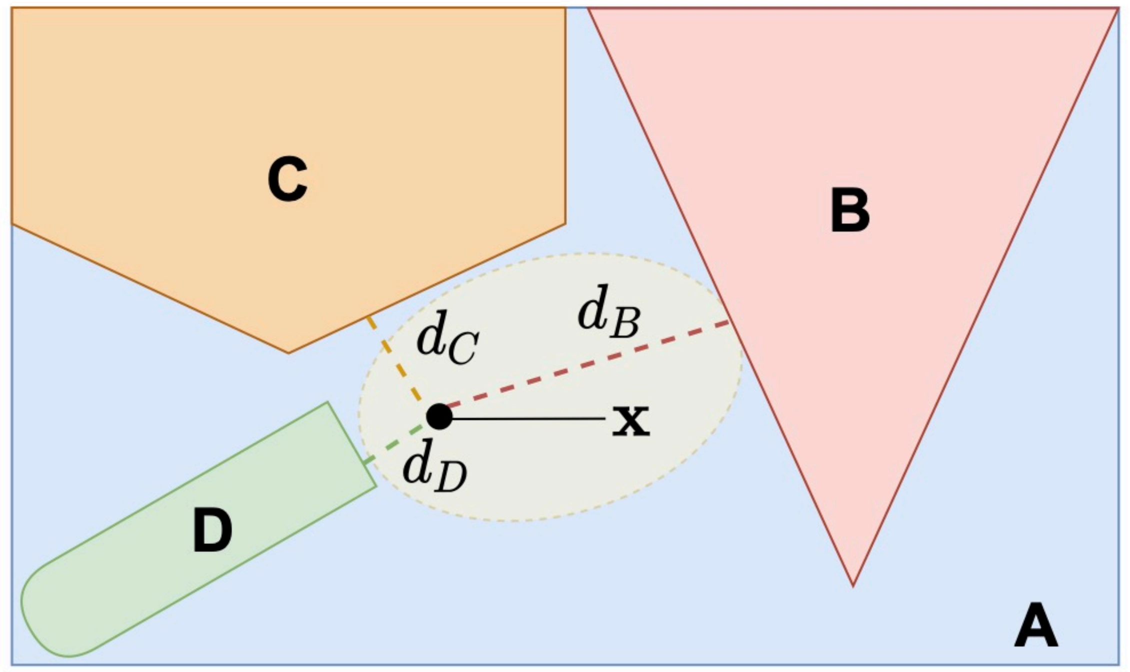
- An adversary $\mathscr{A}_*$ gain access to $S_{\mathscr{V}}$ and train its model $f_{\mathscr{A}_*}$

Data inference

- Motivations:
  - Stolen models are more confident about points in the victim model’s training set than on a random point drawn from task distribution
  - Data trained in the dataset are far from decision boundaries



(a) If $\mathbf{x}$ is in training set



(b) If $\mathbf{x}$ is not in training set

Figure 1: The effect of including $(\mathbf{x}, \text{‘A’})$ in the train set. If $\mathbf{x}$ is in the train set, the classifier will learn to maximize the decision boundary’s distance to $\mathcal{Y} \setminus \{\text{‘A’}\}$. If $\mathbf{x}$ is in the test set, it has no direct impact on the learned landscape.

Data inference

White-box setting

- For any data point $(\mathbf{x_i}, y_i)$, we evaluate its minimum distance Δ to target classes t
  - $\min_{\delta} \Delta(\mathbf{x}, \mathbf{x} + \delta)$ s.t. $f(\mathbf{x} + \delta) = t$

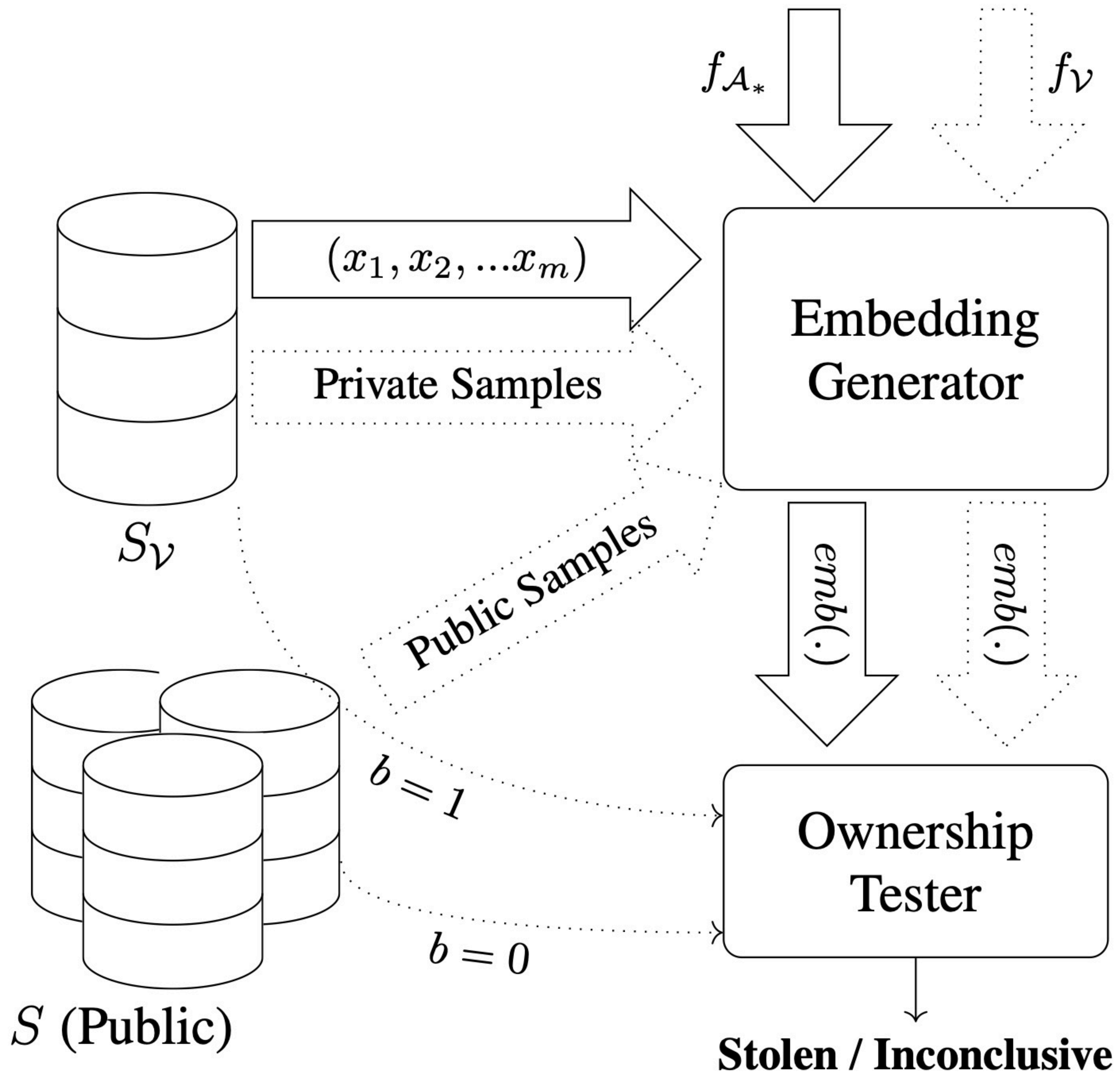


Figure 2: Training (dotted) the confidence regressor with embeddings of public and private data, and victim's model $f_{\mathcal{V}}$; Dataset Inference (solid) using m private samples and adversary model $f_{\mathcal{A}_*}$

Embedding generation

black-box setting

- Starting from an data point $(\mathbf{x_i}, y_i)$, sample with a random direction δ, we take k steps in the same direction until
  - $f(\mathbf{x} + k\delta) = t; t \neq y$

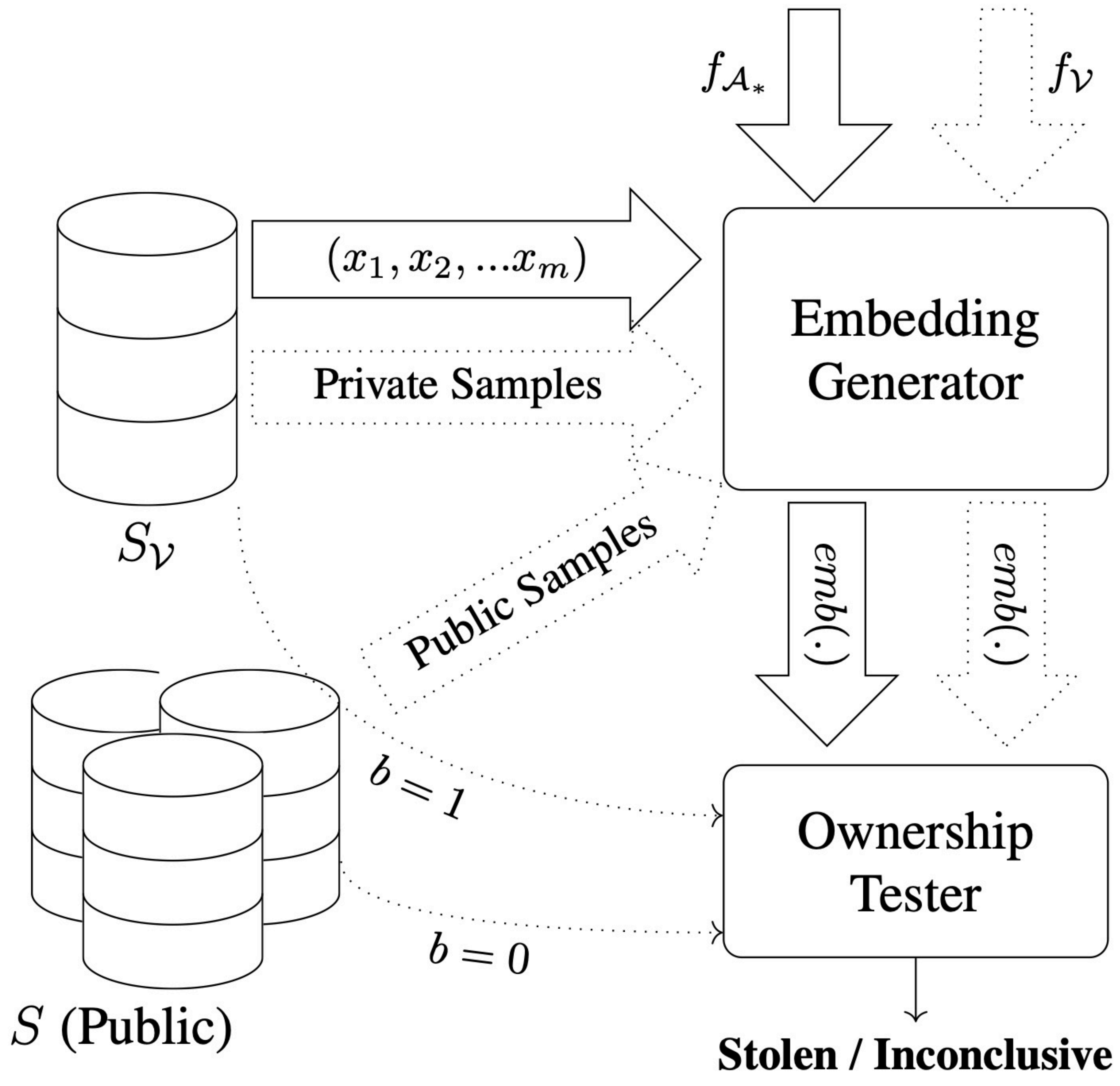


Figure 2: Training (dotted) the confidence regressor with embeddings of public and private data, and victim's model $f_{\mathcal{V}}$; Dataset Inference (solid) using m private samples and adversary model $f_{\mathcal{A}_*}$

Data inference

Confidence regressor

- Min the false positive rate
- Train a regression model $g_{\mathscr{V}}$ -> predict a measure of confidence that it contains the private information

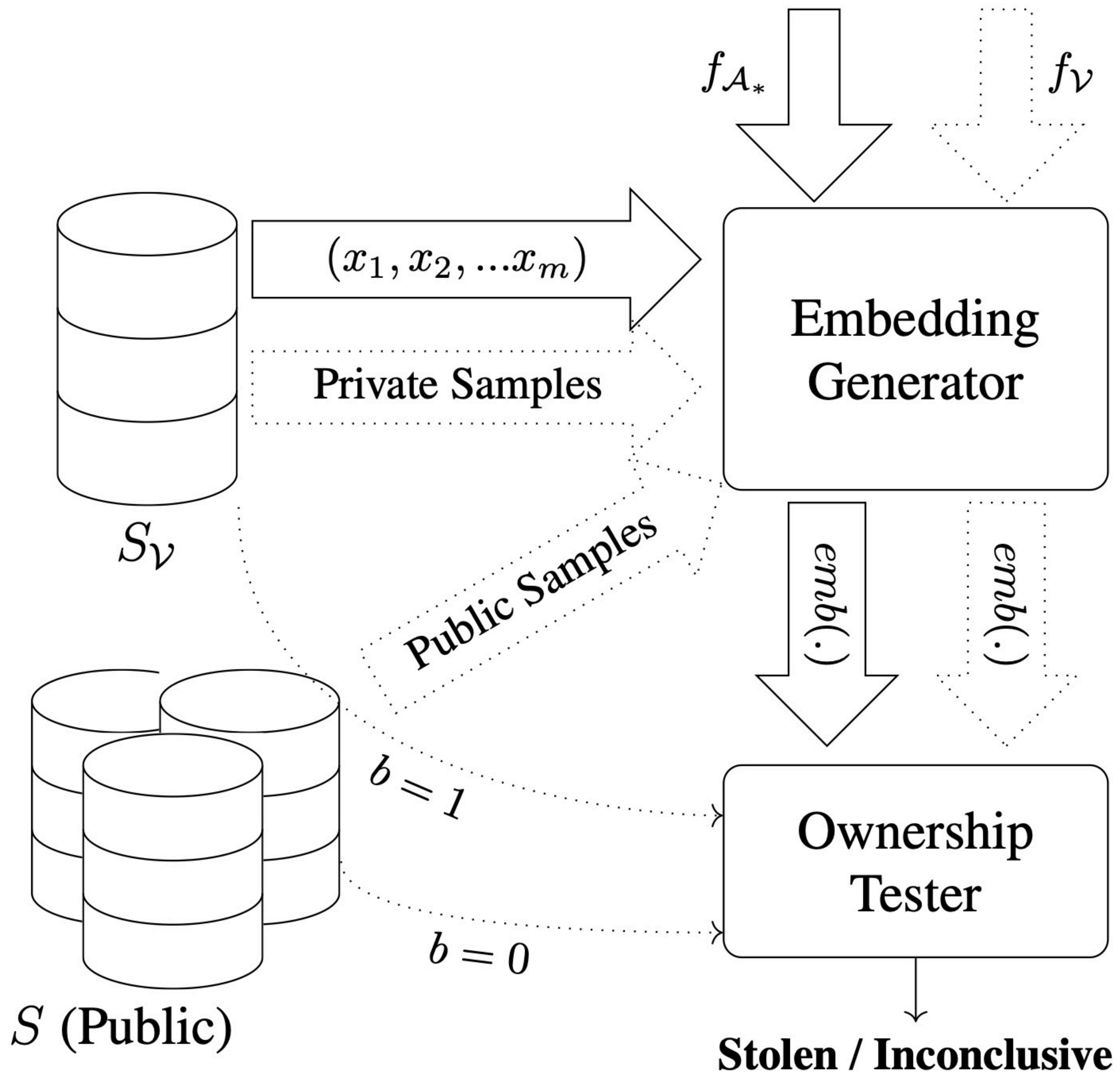


Figure 2: Training (dotted) the confidence regressor with embeddings of public and private data, and victim's model $f_{\mathcal{V}}$; Dataset Inference (solid) using m private samples and adversary model $f_{\mathcal{A}_*}$

Data inference

Hypothesis testing

- Null hypothesis
  - $H_0 : \mu < \mu_\mathcal{V}$ where $\mu = \bar{\boldsymbol{c}}$ and $\mu_\mathcal{V} = \bar{\boldsymbol{c}}_\mathcal{V}$

where $\mu = \bar{\boldsymbol{c}}$ and $\mu_\mathcal{V} = \bar{\boldsymbol{c}}_\mathcal{V}$ are mean confidence scores.

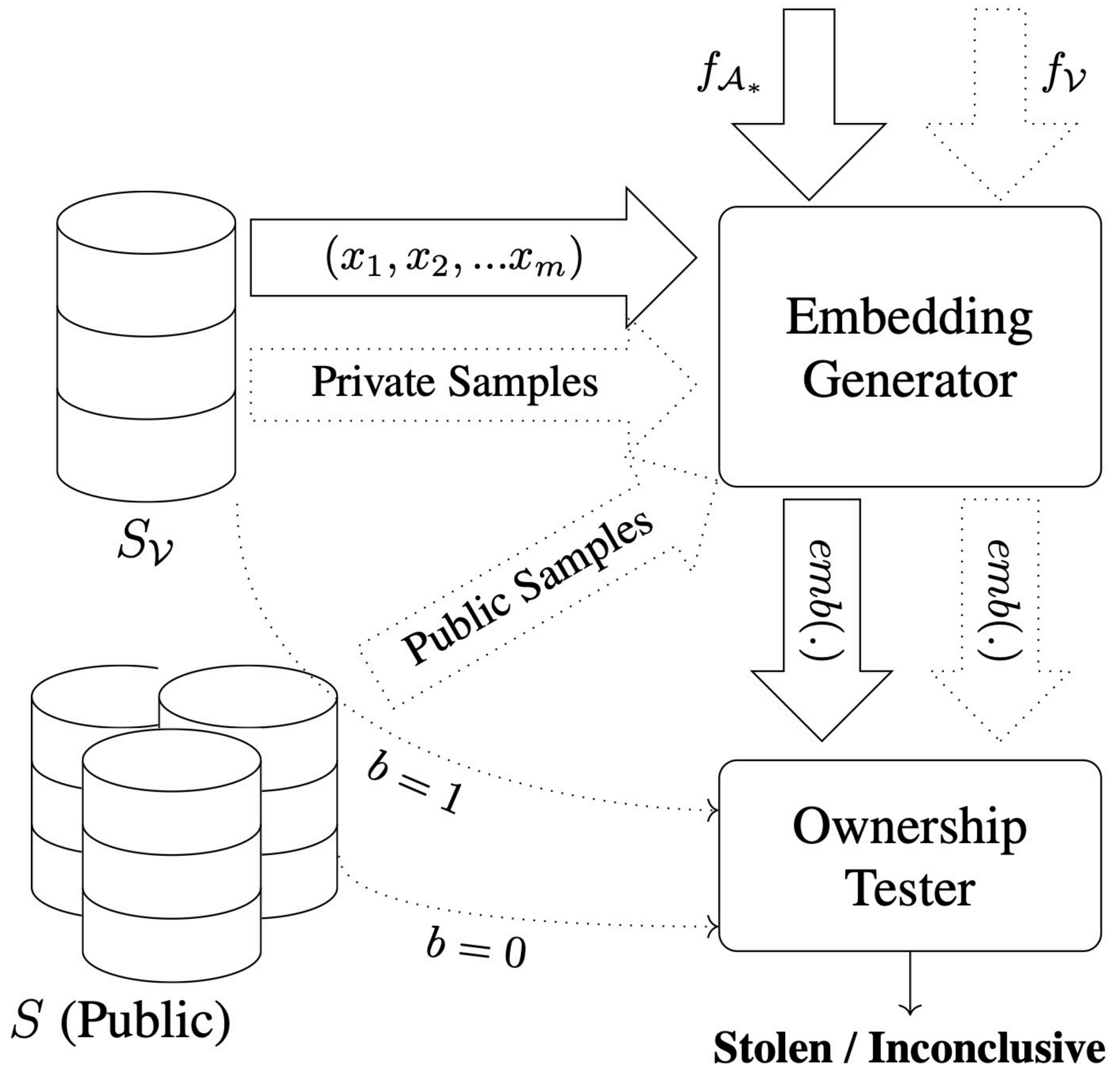


Figure 2: Training (dotted) the confidence regressor with embeddings of public and private data, and victim's model $f_\mathcal{V}$; Dataset Inference (solid) using m private samples and adversary model $f_{\mathcal{A}_*}$

Data inference

Main results

	Model Stealing Attack	CIFAR10				CIFAR100			
		MinGD		Blind Walk		MinGD		Blind Walk	
		$\Delta\mu$	p-value	$\Delta\mu$	p-value	$\Delta\mu$	p-value	$\Delta\mu$	p-value
$\mathcal{V}$	Source	0.838	10^{-4}	1.823	10^{-42}	1.219	10^{-16}	1.967	10^{-44}
$\mathcal{A}_D$	Distillation	0.586	10^{-4}	0.778	10^{-5}	0.362	10^{-2}	1.098	10^{-5}
	Diff. Architecture	0.645	10^{-4}	1.400	10^{-10}	1.016	10^{-6}	**1.471**	10^{-14}
$\mathcal{A}_M$	Zero-Shot Learning	**0.371**	10^{-2}	**0.406**	10^{-2}	0.466	10^{-2}	0.405	10^{-2}
	Fine-tuning	**0.832**	10^{-5}	**1.839**	10^{-27}	**1.047**	10^{-7}	1.423	10^{-10}
$\mathcal{A}_Q$	Label-query	0.475	10^{-3}	1.006	10^{-4}	**0.270**	10^{-2}	**0.107**	10^{-1}
	Logit-query	0.563	10^{-3}	1.048	10^{-4}	0.385	10^{-2}	0.184	10^{-1}
$\mathcal{I}$	Independent	0.103	1	-0.397	0.675	-0.242	0.545	-1.793	1

Table 1: Ownership Tester's effect size (higher is better) and p-value (lower is better) using $m = 10$ samples on multiple threat models (see § 6.1). The highest and lowest effect sizes among the model stealing attacks ($\mathcal{A}_D, \mathcal{A}_M, \mathcal{A}_Q$) are marked in **red** and **blue** respectively.

Data inference

P value

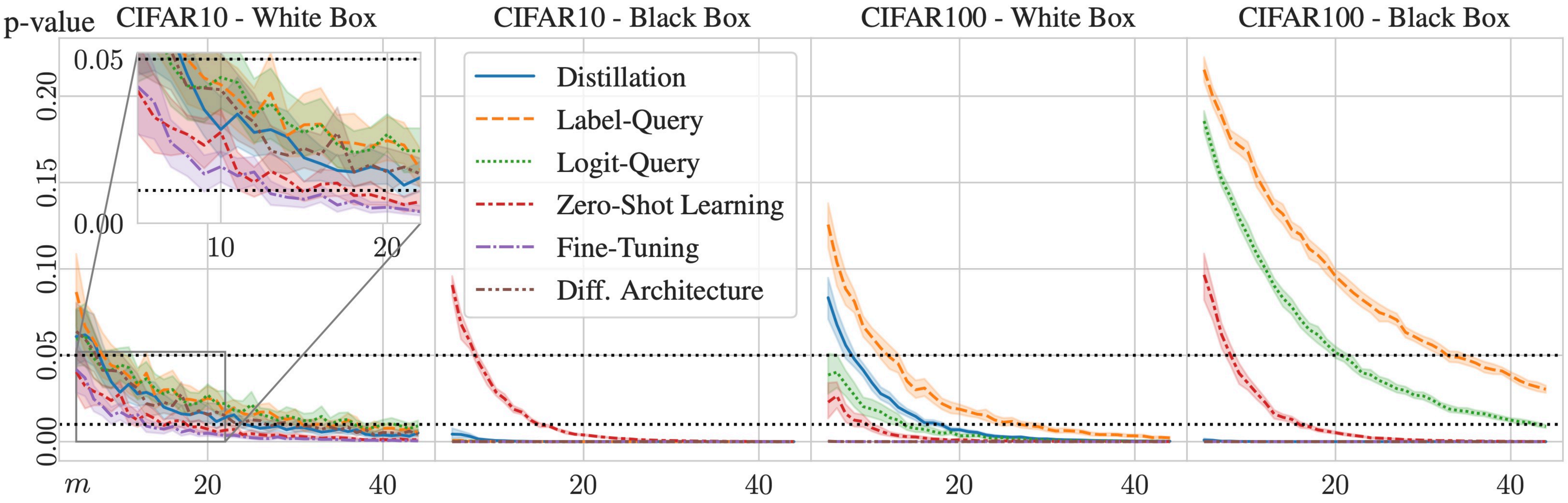


Figure 3: p-value against number of revealed samples (m). Significance levels (FPR) $\alpha = 0.01$ and 0.05 (dotted lines) have been drawn. Under most attack scenarios, the victim $\mathcal{V}$ can dispute the adversary's ownership of $f_{\mathcal{A}_*}$ (with FPR of at most 1%) by revealing fewer than 50 private samples.