Anshuman Suri* and David Evans

Formalizing and Estimating Distribution Inference Risks

Abstract: Distribution inference, sometimes called property inference, infers statistical properties about a training set from access to a model trained on that data. Distribution inference attacks can pose serious risks when models are trained on private data, but are difficult to distinguish from the intrinsic purpose of statistical machine learning—namely, to produce models that capture statistical properties about a distribution. Motivated by Yeom et al.'s membership inference framework, we propose a formal definition of distribution inference attacks general enough to describe a broad class of attacks distinguishing between possible training distributions. We show how our definition captures previous ratio-based inference attacks as well as new kinds of attack including revealing the average node degree or clustering coefficient of training graphs. To understand distribution inference risks, we introduce a metric that quantifies observed leakage by relating it to the leakage that would occur if samples from the training distribution were provided directly to the adversary. We report on a series of experiments across a range of different distributions using both novel black-box attacks and improved versions of the state-of-the-art white-box attacks. Our results show that inexpensive attacks are often as effective as expensive meta-classifier attacks, and that there are surprising asymmetries in the effectiveness of attacks.

Keywords: property inference, distribution inference, privacy-preserving machine learning

DOI 10.56553/popets-2022-0121
Received 2022-02-28; revised 2022-06-15; accepted 2022-06-16.

1 Introduction

Inference attacks aim to infer sensitive information about inputs to a process from its revealed outputs. In machine learning, such attacks usually focus on learning sensitive information about training data from a released model. For example, in a *membership inference* attack [1], the adversary aims to infer whether a particular record was part of the training data. In a *distribution inference* attack, an adversary aims to infer some statistical property of the training dataset, such as the proportion of women in a dataset used to train a smile-detection model [2]. In the research literature, such attacks have previously been called *property inference* and *attribute inference* (confusingly, since this is also used to refer to a type of dataset inference where the adversary infers an unknown sensitive feature of records in the training dataset [3]), and various other terms.

The privacy threat posed by membership inference attacks is well recognized—if an adversary can infer the presence of a particular user record in a training dataset of diabetes patients, it would violate privacy laws limiting medical disclosure. Distribution inference attacks pose a less obvious threat but can also be dangerous. As one example, consider a financial organization that trains a loan scoring model on some of its historical data. An adversary may use a distribution inference attack to infer the proportion of the training data having a specific value for some protected attribute (e.g., race), which might be a sensitive property of the training data. Distribution inference attacks can also pose a threat to distributed training: curious users wanting to learn sensitive information about training distributions of fellow participants, which are competitors in settings like cross-silo federated learning [4], thus leaking sensitive information. Other examples include inferring the sentiment of emails in a company from a spam classifier, or inferring the volume of transactions from fraud detection systems [5]. We also demonstrate examples where adversaries learn properties of graphs used in training, such as accurately estimating their average clustering coefficient. Inferring statistical properties of a training distribution can also be used to enhance membership inference attacks or to reveal bias in the training data [6].

Previous works have used several different informal notions of distribution inference attacks ([7–9]), but there is no established general formal definition. In this work, we formalize distribution inference attacks based on a critical insight: the key difference between these attacks and other inference attacks is: the adversary's goal

***Corresponding Author: Anshuman Suri:** University of Virginia, E-mail: anshuman@virginia.edu
David Evans: University of Virginia, E-mail: evans@virginia.edu

in the former is to learn about the training *distribution*, not about the specific training *dataset*. Dataset inference attacks, such as membership inference [1], attribute inference [3], and ownership-resolution [10] operate on the level of training records. Attacks like membership inference are connected to differential privacy which bounds the ability to distinguish neighboring datasets. By contrast, distribution inference attacks attempt to learn statistical properties of the underlying distribution from which training data is sampled. A formal definition and a clear threat model can be useful in several ways—quantifying information leakage, assessing and comparing the practicality of threat models, and drawing possible links between distribution inference risk and other properties such as robustness and fairness.

Contributions. We provide a general and straightforward formalization of distribution inference attacks as a cryptographic game (Section 3), inspired by Yeom *et al.*'s membership inference definition [11]. Our definition is generic enough to capture a variety of statistical properties of the underlying distribution including, but not limited to, the attribute ratios that have been the focus of previous property inference attacks. We define an intuitive metric for measuring the amount of leakage observed in a simulated attack, along with theorems that compute this metric for the case of ratios of binary functions, regressions that estimate the proportion of training data having a given attribute, and degree distributions (for graphs) as properties (Section 4).

We report on experiments evaluating distribution inference as a risk across several datasets and properties, providing the first systematic evaluation of how inference risks vary as distributions diverge. Studying patterns in distribution inference risk and how they correlate with underlying properties helps compare trends across experiments , helping identify datasets and models that are highly sensitive to such inference attacks. To conduct our experiments we introduce two simple black-box distribution inference attacks, and extend the white-box permutation-invariant network [12], the current state-of-the-art property inference method, to add support for convolutional layers (Section 5). This enables us to conduct distribution inference attacks on deep neural networks, even when target models are trained from scratch. Section 6 reports results from our experiments with these attacks on a variety of datasets, tasks, and inference goals, revealing that simple black-box attacks often perform surprisingly well, and in some settings white-box meta-classifier attacks can reveal information comparable to sampling dozens of records from the training distribution. We analyze the information gleaned up by meta-classifiers across layers, finding that most information can be find in just one or a few layers, enabling less expensive meta-classifier attacks (Section B).

2 Previous Work

This section summarizes work on formal definitions of privacy, distribution inference attacks and defenses.

Privacy Definitions. Most formal privacy definitions, including numerous variations on differential privacy [13], focus on bounding inferences about specific data elements, not the statistical properties of a dataset. The key privacy notion of traditional differential privacy is intuitively connected to the risk to an individual in contributing their data to a dataset — this corresponds well to dataset privacy risks, but does not capture distribution inference risks; indeed, the main goal of most differentially private mechanisms is to satisfy the inference bound for individual data while providing the most accurate aggregate statistics possible. One notable exception is the Pufferfish framework [14], which introduces notions that allow capturing aggregates of records via explicit specifications of potential secrets (e.g., distribution of vehicle routes in a shipping company) and their relations. Zhang et al. extend the Pufferfish framework to define the concept of "attribute privacy" [15], including a notion of distributional attribute privacy that takes a hierarchical approach for parameterizing distributions and could be instantiated to capture notions of distribution inference such as the fraction of records with some attribute. Although these definitions are promising and valuable, none of them satisfy our simple goal to define distribution inference attacks in a way that is general and powerful, while clearly distinguishing inferences that are considered attacks from allowable statistical inferences.

A recent attempt to formalize property inference [5] consists of a framework that reduces property inference to Boolean functions of individual members, posing the ratio of dataset members satisfying the given function as the property. These ratio-based formulations limit the kinds of distribution inferences considered since they cannot capture many other kinds of statistical properties of the training distribution that may be sensitive, like the degree distribution of a graph [16]. Ratio-based formulations assume the property function is applicable over individual data points, while for graphs it is an aggregation over interconnected nodes.

Distribution Inference Attacks. All previous distribution inference attacks in the literature take a meta-classifier approach—the adversary trains models on datasets with different properties, then trains a meta-classifier using those models. The adversary then uses the meta-classifier to predict a property of the victim's training data, which is usually related to the ratio of members satisfying some Boolean property. Ateniese et al. [2] were the first to identify the threat of distribution inference (termed *property inference*) and proposed a meta-classifier attack targeting Support Vector Machines and Hidden Markov Models. The proposed attacks can successfully infer the accent of speakers in speech-to-text systems, or presence of particular traffic in network traffic classification systems. Model representations for training the meta-classifier can take several forms: using model weights [12], or activations/logits for a set of query points [17]. These methods show promise, achieving better-than-random results for several properties, tasks, and models across different domains. For instance, predicting a doctor's specialty based on rating-prediction systems on text reviews [9], identifying accents of speakers in voice-recognition models [2], and even predicting if a model has been trained with Trojans [17]. Although these approaches achieve high accuracies on "toy-like" classifiers like decision trees and shallow neural networks, and distributions that are highly disparate, successful property inference attacks have not been demonstrated on realistic (or even semi-realistic) deep neural networks or complex datasets. Our work is the first to demonstrate the capability of such attacks to work on large convolutional networks and different datasets across domains.

Zhou et al. [6] extend distribution inference attacks to directly infer property ratios for Generative Adversarial Networks (GANs). In their attack, the adversary trains shadow GANs and launches a black-box attack on the victim model using generated samples. Although their attack setting includes targeting large GAN models on complex datasets like CelebA [18], the adversary does not use the victim model's model parameters directly as is done by our extension to convolutional models (Section 5.2.2). Similarly, Pasquini et al. [19] extend distribution inference attacks to a split-learning setting and only target parts of the victim model. Zhang et al. extended this approach to graphs and their properties [20], distinguishing training graphs according to properties such as the number of nodes and edges using attacks that assume access to graph embeddings.

Defences Against Distribution Inference. Unlike other notions of privacy like membership privacy, there are no known defenses against distribution inference. Differential privacy does not mitigate distribution inference risks since it obfuscates the contribution of individual records, while the adversary in our setting cares about statistical properties of the underlying distribution. Attempts in previous works to evaluate differential privacy as a potential defense [2] show that it does not work. Removing sensitive attributes from datasets is not an effective defense either [9], since correlations between attributes still leak information. Using node multiplicative transformations [12] has been proposed as a defense for neural networks with ReLU activations, but provides no protection against black-box attacks.

3 Distribution Inference

Threat Model. We model the adversary's knowledge of the underlying distribution through data sampled from that distribution. For complex, high-dimensional non-synthetic data, this is usually the only way to capture knowledge of a data distribution. While the threat model focuses on distributions, we use non-overlapping sampled data to empirically model those distributions.

A natural extension of our threat model incorporates a poisoning opportunity where the adversary participates in the training process. Such an adversary may poison the training dataset by injecting adversarially crafted datapoints [5] or control the training procedure itself to introduce backdoors. Recent works look at a similar scenario where the adversary participates in the learning process via federated-learning, launching attribute-reconstruction attacks using epoch-averaged model gradients [21]. In this paper, we only consider adversaries with no ability to observe or influence the training process.

Setup. Let $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ be a public distribution between data, $\mathcal{X}$, and its corresponding labels, $\mathcal{Y}$. We assume both the model trainer $\mathcal{T}$ and adversary $\mathcal{A}$ have access to $\mathcal{D}$. Both parties also have access to two functions, $\mathcal{G}_0$ and $\mathcal{G}_1$, that transform distributions. Inferring properties of the distribution can reveal sensitive information in many scenarios, which can be captured by suitable choices of $\mathcal{G}_0$ and $\mathcal{G}_1$. Using such functions along with the underlying distribution $\mathcal{D}$ makes the setup less restrictive than defining two arbitrary distributions—since the functions $\mathcal{G}_0, \mathcal{G}_1$, and $\mathcal{D}$ are considered public knowledge, anyone can recreate these distributions. Using functions on the same distribution $\mathcal{D}$ emphasizes the fact that these two distributions stem from the same underlying distribution $\mathcal{D}$, enabling the definition to capture a wide class of possible attack goals and scenarios by selecting appropriate functions for $\mathcal{G}_0$ and $\mathcal{G}_1$.

To illustrate these definitions, we present a concrete example inspired by Chase et al. [5]. Let $\mathcal{D}$ be a distribution of emails with labels for spam/ham. $\mathcal{G}_0$ is applied over $\mathcal{D}$ to yield a modified distribution $\mathcal{G}_0(\mathcal{D})$ with 0.8 probability of sampling an email with negative sentiment, *i.e.* a dataset sampled uniformly at random from this distribution would have 80% of the emails with negative sentiment. Similarly, $\mathcal{G}_1(\mathcal{D})$ could be another distribution with this probability as 0.5. The adversary thus wants to know if the training distribution is biased, which, if inferred near the financial quarter, can be used to predict if the company is under-performing. Alternatively, an ambitious adversary could even consider directly inferring this proportion (which was not considered in Chase et al. [5], but we explore in Theorem 4.3 and our regression experiments on other datasets).

We propose a general and straightforward experiment to formalize property inference attacks, inspired by Yeom *et al.*'s cryptographic game definition of membership inference [11]. $\mathcal{T}$ picks one of the $\mathcal{G}_{\{0,1\}}$ distribution transformers at random and samples a dataset S from the resulting distribution. Given access to a model M trained on S, the adversary aims to infer which of the two distribution mappers was used:

	Trainer $\mathcal{T}$		**Adversary $\mathcal{A}$**
1 :	$b \leftarrow\!\$\ \{0,1\}$		
2 :	$S \sim \mathcal{G}_b(\mathcal{D})$		
3 :	$M \xleftarrow{train} S$		
4 :		$\xrightarrow{M}$	
5 :			$\hat{b} = \mathcal{H}(M)$

In this work, we assume the adversary has no control over the training process (Step 3). For cases where the adversary has access to the training data, it can trivially infer desired properties by inspecting it. Our definition could be adapted to other scenarios such as federated learning [22] by providing additional information to the adversary or allowing the adversary to have some control over S or other aspects of the training process, but we do not consider such settings in this paper.

If $\mathcal{A}$ can successfully predict b via $\hat{b}$, then it can determine which of the training distributions was used. The advantage of the adversary $\mathcal{A}$ using algorithm $\mathcal{H}$ is:

$$\mathrm{Adv}_{\mathcal{H}} = \left| \Pr\left[\hat{b} \;\middle|\; b\right] - \Pr\left[\hat{b} \;\middle|\; \neg\, b\right] \right| .$$

This advantage is negligible when the adversary does no better than random guessing. We do not assume the adversary knows how training data is collected, or has any access to it: just that it knows the underlying common distribution $\mathcal{D}$, and has a goal of distinguishing between sub-distributions $\mathcal{G}_0(\mathcal{D}), \mathcal{G}_1(\mathcal{D})$ of that distribution. The adversary does not need to know the actual training distribution—indeed, learning about this is the goal of the attack. They just need to have hypotheses worth testing, and thus $\mathcal{G}_0$ and $\mathcal{G}_1$ are defined by the adversary based on what they want to test.

Limitations of the Definition. This definition is simple and general, but does not capture all kinds of distribution inference attacks. It assumes a setting where the adversary attempts to distinguish between two particular distributions defined by the adversary. Multiple experiments could extend the definition to a set of distributions, but the definition does not directly capture the regression attacks we demonstrate where the adversary estimates what proportion of a training dataset has a given property directly. Our definition also assumes the adversary has prior knowledge of the two sub-distributions to distinguish. Some knowledge of the statistical property the adversary wants to learn about the victim's training distribution is inherent in the nature of a distribution inference attack, but this may not always be in the form of knowledge of possible distributions. In our experiments, we model an adversary's knowledge of the underlying distribution through a sampled, non-overlapping dataset, which the adversary then uses to construct approximations of different sub-distributions.

Applying the Definition. Seminal works on property inference [2, 12] involve a model trained either on the original dataset or a version modified to be biased towards some chosen attribute. Our definition can be used to describe these attacks by setting $\mathcal{G}_0$ to the identity function (so $\mathcal{G}_0(\mathcal{D})$ is original distribution $\mathcal{D}$) and $\mathcal{G}_1$ to a filter that adjusts the distribution to have a specified ratio over the desired attribute. With respect to a binary property function, $f : \mathcal{X} \rightarrow \{0, 1\}$, $\mathcal{D}$ can be characterized using a generative P.D.F:

$$\rho_{\mathcal{D}}(\mathbf{x}) = \sum_{c \in \{0,1\}} p(c) \cdot p(\mathbf{x} \mid c), \tag{1}$$

where $p(c)$ is a multinomial distribution representing the probabilities over the desired (binary) property function f and its possible values c, and $p(\mathbf{x} \mid c)$ is the generative conditional probability density function. Then, $\mathcal{G}_1(\mathcal{D})$ can be expressed using the following probability density function, with a prior $\hat{p}$:

$$\rho_{G_1(\mathcal{D})}(\mathbf{x}) = \sum_{c \in \{0,1\}} \hat{p}(c) \cdot p(\mathbf{x} \mid c) \tag{2}$$

$$\hat{p}(1) = \alpha,\ \hat{p}(0) = 1 - \alpha \tag{3}$$

where α is the probability of a randomly sampled point satisfying the property function f. Thus, a uniformly randomly sampled dataset from $\mathcal{G}_1(\mathcal{D})$ would have an expected ratio of α of its members satisfying f. Additionally, we can modify $\mathcal{G}_0$ with a similarly adjusted prior, enabling the adversary to distinguish between any two arbitrary ratios [9]. In fact, our definition subsumes the one proposed by Chase *et al.* [5] for the case of ratios over Boolean functions via the following instantiation:

$$\begin{aligned} D_-,\ D_+ &= \mathcal{G}_0(\mathcal{D}),\ \mathcal{G}_1(\mathcal{D}) \\ t_0,\ t_1 &= \alpha_0,\ \alpha_1, \end{aligned} \tag{4}$$

along with setting $c = f(x)$ in Equation 3.

Our definition, however, is not limited to describing proportional properties. For example, it can also be used to define the distributions over degrees for graph-based datasets, and infer properties of the underlying degree distributions. For this case, we represent graphs as samples from degree distributions: data with different degrees is sampled and then combined together in one graph. Since the adversary only cares about properties pertaining to the degrees of nodes (and not their attributes or other characteristics), it is safe to represent these samples purely in terms of degree distributions. This can then be used to target properties such as the mean-node degree of graphs, as we show in Section 6.3.

4 Quantifying Leakage

Assessing the power of an attack is important for understanding it scientifically, and can also be of practical importance for both the victim and the adversary. Consider the most explored case in the literature—ratios of members satisfying a Boolean function. Intuition suggests that distributions with more different ratios (e.g., 0.2 and 0.9) would be easier to distinguish than more similar ones (e.g., 0.2 and 0.3), and most previous distributions inference results have focused on highly disparate distributions (often only showing meaningful distinguishing power when one of the ratios is at a 0.0 or 1.0 extreme).

Our framework enables us to quantify the amount of leakage observed in an attack by relating what an adversary is able to learn from a disclosed model to what they would learn from directly sampling examples from the training distribution. As a setup, we provide the following lemma, proven in Appendix A.1, that gives an upper bound on the distinguishing accuracy (which we define as the probability of an adversary correctly inferring the underlying training distribution of a model) of any statistical test distinguishing between two distributions that differ in the proportion of records satisfying some Boolean property, using n samples:

Lemma 4.1. Given two Boolean-property proportional distributions $\mathcal{G}_0(\mathcal{D}), \mathcal{G}_1(\mathcal{D})$ with proportion values α_0, α_1 derived from the same underlying distribution $\mathcal{D}$, the distinguishing accuracy between models trained on datasets of size n from these distributions is at most

$$\frac{1}{2} + \frac{\min\left\{\sqrt{1 - \left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)^n}, \sqrt{1 - \left(\frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right)^n}\right\}}{2}.$$

First, we consider the most powerful possible adversary as one that can perfectly reconstruct training records from a model. The most leaked to such an adversary is a perfect reconstruction of all the training data. Of course, we do not expect an adversary to reconstruct the training dataset fully, and an adversary can succeed in a high confidence distribution inference attack without being able to reconstruct any training records perfectly. Such a perspective is useful, though, for quantifying the power of an attack in a way that allows comparisons between attacks distinguishing distributions with different levels of variation. For some observed performance ω via an attack, we can compute the corresponding value of n that would give an upper bound on accuracy as ω. This value of n, which we term as n_{leaked}, thus quantifies the size of the dataset "leaked" by the attack. In other words, it is equivalent to the adversary being able to draw n_{leaked} samples from the training distribution and executing an optimal distinguishing statistical test. The following theorem, proven in Appendix A.2, shows how to compute n_{leaked} for an observed attack for the kind of distributions described above.

Theorem 4.2. Given two Boolean-property proportional distributions $\mathcal{G}_0(\mathcal{D}), \mathcal{G}_1(\mathcal{D})$ with proportion values α_0, α_1 derived from the same underlying distribution $\mathcal{D}$, and distinguishing accuracy ω using some attack,

$$n_{\text{leaked}} = \frac{\log(4\omega(1-\omega))}{\log(\max\left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}, \frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right))}.$$

A high value of n_{leaked} means the adversary is learning a lot about the underlying distribution, just using the given model. It helps put the attack's strength in perspective, given how similar the two distributions are. For instance, distinguishing between $\alpha_0 = 0.5$ and $\alpha_1 = 1.0$ with distinguishing accuracy $\omega = 0.95$ corresponds to $n_{\text{leaked}} \approx 3$, whereas distinguishing between $\alpha_0 = 0.5$ and $\alpha_1 = 0.52$ with the same accuracy would correspond to $n_{\text{leaked}} \approx 42$. This aligns with intuition: the latter distribution is more similar and thus, should be "harder" for an attack to achieve the same kind of performance, and this notion is exactly what n_{leaked} aims to capture.

Note that this analysis is based on modeling an "optimal attack" where the adversary is able to directly sample records from the training distribution. This is just for deriving an expression for n_{leaked}, and not meant to assume any such attack. The expression above (and subsequent expressions for n_{leaked}) is a useful measure of an attack's effectiveness that quantifies the leakage observed in the attack by relating it to the amount of information leaked in an "optimal attack" where the adversary samples training distribution records directly rather than inferring properties from a revealed model.

Regression over α. The case of distinguishing between ratios can be further extended to consider an adversary that wishes to directly predict the proportion value α for a given distribution. The following theorem, proven in Appendix A.3, shows how to compute n_{leaked} for an observed attack with square error ω.

Theorem 4.3. Given a Boolean-property proportional distribution with proportion value α, and square error ω observed using some attack,

$$n_{\text{leaked}} = \frac{\alpha(1-\alpha)}{\omega}.$$

This extends our notion of n_{leaked} to adversaries that directly infer the underlying ratio α, a more realistic adversary goal considered in some of our experiments.

Graphs as Distributions of Natural Numbers. Similar to the ratio case, our framework enables us to compute n_{leaked} when working with distributions of natural numbers. For the purpose of distinguishing between graph distributions, this notion of 'distribution of numbers' can be extended to graphs by studying their degree distributions. As a setup, we provide the following lemma, proven in Appendix A.4, that gives an upper bound on the distinguishing accuracy of any statistical test distinguishing between two distributions following Zipf's law, using n samples:

Lemma 4.4. Given two distributions of natural numbers, $\mathcal{G}_0(\mathcal{D})$ and $\mathcal{G}_1(\mathcal{D})$ that follow Zipf's law, with N_0 and N_1 elements (without loss of generality assume $N_0 \leq N_1$) and parameters s_0, s_1 respectively, the distinguishing accuracy between models trained on graphs with n nodes from these distributions is at most:

$$\frac{1}{2} + \frac{\sqrt{1 - \left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}} N_0^{(s_0-s_1)\mathbb{I}[s_1>s_0]}\right)^n}}{2}.$$

Here, $H_{n,s}$ is the n^{th} generalized Harmonic number of order s, $H_{n,s} = \sum_{k=1}^{N} k^{-s}$. Together, these two parameters are related to the expected mean of the distribution α_b (mean node-degree, for degree distributions) as:

$$\alpha_b = \frac{H_{N_b,s_b-1}}{H_{N_b,s_b}} \tag{5}$$

Similar to the case of different Boolean-property ratios distributions, we can compute n_{leaked} for a given attack and its observed performance (proof in Appendix A.5):

Theorem 4.5. Given two distributions of natural numbers distributions $\mathcal{G}_0(\mathcal{D})$, $\mathcal{G}_1(\mathcal{D})$ that follow Zipf's law, with N_0, N_1 elements (without loss of generality assume $N_0 \leq N_1$) and parameters s_0, s_1 respectively, and observed distinguishing accuracy ω,

$$n_{\text{leaked}} = \frac{\log(4\omega(1-\omega))}{\log\left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}}\right) + (s_0 - s_1)\mathbb{I}[s_1 > s_0]\log(N_0)}.$$

Compared to the ratio distinguishing attacks, attacks on the ogbn-arxiv dataset are much more successful: reaching near-perfect distinguishing accuracies as well as the highest n_{leaked} numbers (Table 1).

5 Attacks

In our experiments, we use two simple black-box attacks and the state-of-the-art white-box meta-classifier attack. We extend the meta-classifier attack to support convolutional neural networks (Section 5.2.2). These attacks do not assume anything about the underlying distributions other than the availability of the underlying distribution and knowledge of the public $\mathcal{G}_0$ and $\mathcal{G}_1$ transformers.

5.1 Black-Box Attacks

Black-box attacks assume the adversary has access to representative data for distributions, and API access to the target model which outputs its predicted label for a submitted input. Although access to model parameters is unavailable in this setting, its model-agnostic nature allows launching attacks on any target model.

5.1.1 Loss Test

A simple algorithm $\mathcal{H}$ is to test the accuracy of the model on datasets from the two candidate distributions, and conclude that the training distribution is closest to whichever test dataset the model performs better on. For data samples $S_{b\in\{0,1\}} \sim \mathcal{G}_b(\mathcal{D})$:

$$\hat{b} = \mathbb{I}[acc(M, S_0) < acc(M, S_1)], \tag{6}$$

where $acc(M, S)$ is the accuracy of model M on some data S, and $\mathbb{I}$ is the indicator function. Intuitively, a model would have higher accuracy on data sampled from the training distribution, compared to another distribution. This method does not require the adversary to train models, but only to have access to suitable test distributions and the ability to submit samples to the target model. The data held by the adversary here is not overlapping with the data used by the victim to train its models, ruling out any potential for leakage via shared data. Although we use accuracy in Equation 6, the adversary can use any metric.

5.1.2 Threshold Test

The Loss Test assumption may not hold for some pairs of distributions if one distribution is inherently easier to classify than the other. To account for this, we consider an attack where the adversary trains and uses a small (balanced) sample of models from each distribution to identify which of S_0 or S_1 maximizes the performance gap between its models.

$$\begin{aligned}\gamma_{c\in\{0,1\}} &= \sum_{i;y_i=0} acc(M^i, S_c) - \sum_{i;y_i=1} acc(M^i, S_c)\\ k &= \mathbb{I}[|\gamma_0| < |\gamma_1|],\end{aligned} \tag{7}$$

where $y_i \in \{0, 1\}$ denotes that the model M^i is trained on a dataset sampled from $\mathcal{G}_{\{0,1\}}(\mathcal{D})$ respectively. After identifying $k_{\in\{0,1\}}$, the adversary derives a threshold λ to maximize accuracy distinguishing between models trained on datasets from the two distributions (using a simple linear search). Assuming γ_k is positive,

$$\begin{aligned}\delta(\Lambda) = &\sum_{i;y_i=0} \mathbb{I}[acc(M^i, S_k) \geq \Lambda] + \\ &\sum_{i;y_i=1} \mathbb{I}[acc(M^i, S_k) < \Lambda]\\ \lambda = &\arg\max_{\Lambda} \delta(\Lambda).\end{aligned} \tag{8}$$

The adversary then predicts $\hat{b} = \mathbb{I}[acc(M, S_k) \geq \lambda]$ (or with a $<$ inequality when $\gamma_k < 0$). Thus, the adversary uses a sample of local models to derive a classification rule, which it then uses to infer the training distribution of the target model. For the same reasons as Loss Test the data used by the adversary here, for both training its local set of models and computing the threshold), is non-overlapping with the victim's data.

5.2 White-Box Attacks

White-box attacks assume the adversary has full access (parameters) to the trained model and a large enough amount of representative data to train local models on the candidate distributions.

5.2.1 Meta-Classifiers

The state-of-the-art property inference attack uses Permutation-Invariant Networks as meta-classifiers [12]. The meta-classifiers take as input model parameters (weights, bias) and predict the training distribution directly. This architecture is designed to be invariant to neuron orderings inside neural network layers, which it achieves by utilizing the DeepSets [23] architecture. Neuron-ordering invariance is achieved via a set of transforming functions, ϕ_i (for each layer i), over each row of the layer weight matrix. The outputs of these functions are then summed to create a layer representation L^i, thus achieving invariance to the ordering of the neurons within each layer. Since the meta-classifier is itself a classifier that requires many models (800 per distribution Ganju et al.'s work [12]) trained on the two distributions for training, this attack is only feasible for adversaries with access to sufficient data from both training distributions and considerable computational resources.

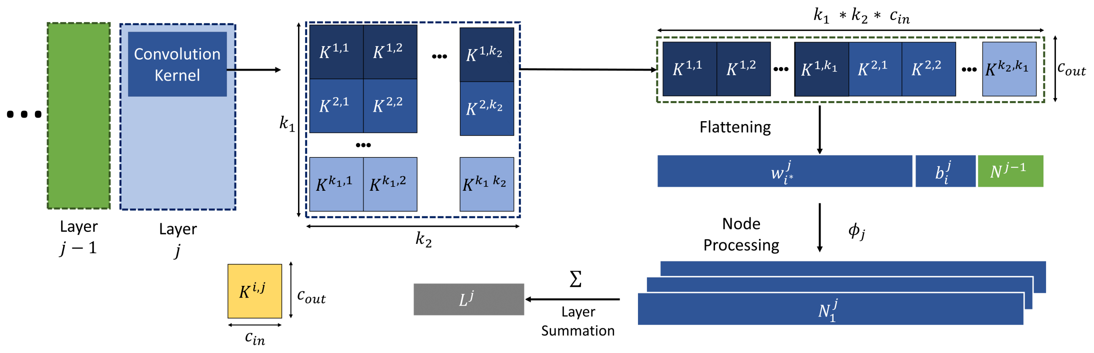


Fig. 1. Transforming a $k_1 \times k_2$ kernel matrix K (with input channels c_{in} and output channels c_{out}) into a 2-dimensional weight matrix for compatibility with the Permutation Invariant Network architecture. The node-processing functions ϕ_i and the rest of the pipeline are identical to the Permutation Invariant Network described in Section 5.2.1.

5.2.2 Targeting Convolutional Neural Networks

The Permutation-Invariant Network only supports linear layers in a feed-forward architecture; previous work only considered two or three layer MLPs on small datasets [12], or single-layer recurrent neural networks [9]. Applying the same architecture on top of convolutional layers requires adaptation since kernel matrices are four-dimensional. While there is permutation invariance within channels, the kernel itself is sensitive to permutations by nature of how convolution operations work. We extend property inference attacks to convolutional networks and demonstrate their effectiveness on models with up to eight layers, trained from scratch.

Figure 1 illustrates our method. Let K be a kernel of size (k_1, k_2) associated with some convolutional layer, with input and output channel dimensions c_{in} and c_{out} respectively. While designing the architecture to capture invariance, it is important to remember that unlike neurons in linear layers, positional information in the kernel matters. Thus, any attempt to capture invariance should be limited to the mapping between input and output channels of a convolutional kernel. We flatten the kernel of size $(k_1, k_2, c_{in}, c_{out})$ such that the resulting matrix is of size $(k_1 \times k_2 \times c_{in}, c_{out})$. Concatenating along the input channel dimension helps preserve location-specific information learned by the kernel while capturing permutation invariance across channels.

This two-dimensional matrix is then processed like linear layers are in Permutation-Invariant Networks (using the same notation as Section 5.2.1), applying function ϕ_i and summing to capture invariance while concatenating feature representations from prior layers. Like the original architecture, the bias component can be concatenated to the kernel matrix itself. Since this feature extraction process on a convolutional layer also produces a layer representation, it can be easily incorporated into the existing architecture to work on models with a combination of convolutional and linear layers.

6 Experiments

To better understand the risks of distribution inference, we execute distribution inference attacks to measure their ability to distinguish distributions with varying disparity on tabular, image, and graph datasets. Although it is unknown how close these attacks are to the best possible distribution inference attacks, they help demonstrate an empirical lower bound on adversarial capabilities and can be helpful in estimating general trends. Code for reproducing our experiments will be publicly released and is available for reviewers in an anonymized repository: https://github.com/iamgroot42/FormEstDistRisks.

6.1 Datasets

We report on experiments using five datasets, summarized in Table 1. We construct non-overlapping data splits between the simulated adversary, $\mathcal{A}$, and model trainer $\mathcal{T}$. These non-overlapping splits help better capture a realistic scenario where the adversary has access to training data from the distribution $\mathcal{D}$ but is unlikely to have any of the model trainer's data (which is considered private). Both parties then modify their data to emulate a distribution property, and then sample training datasets from these adjusted distributions to train their models. This sampling, along with the dis-

Dataset	Task	Property	Size		Binary		Regression
			Adversary	Victim	n_{leaked}	$\|\alpha_0 - \alpha_1\|_{0.75}$	n_{leaked}
Census	Income prediction	Ratio of females	3200	3200	0.2	0.5	8.8
		Ratio of whites	1800	1800	0.1	0.6	6.0
CelebA	Smile identification	Ratio of females	22,000	36,000	0.3	0.5	10.6
	Gender prediction	Ratio of young people	6700	18,800	0.2	0.6	5.1
RSNA Bone Age	Age prediction	Ratio of females	1800	3600	6.2	0.2	269.4
ogbn-arxiv	Node classification	Mean node-degree	40,000	97,000	30.6	-	-
Chord		Average clustering coefficient	135,216	135,523	-	-	-

Table 1. Descriptions of datasets, along with the effectiveness of inference attacks (best of all Loss Test, Threshold Test, and meta-classifier for binary classification, and meta-classifier for regression) while varying ratios of distributions. $\|\alpha_0 - \alpha_1\|_{0.75}$ is the minimum difference in ratios observed that has at least 75% average accuracy. For binary classification, n_{leaked} is the median effective n value based on Theorem 4.2 using the maximum distinguishing accuracies across all experiments and pairs of property ratios (degrees in the case of ogbn-arxiv) without outliers. For regression, n_{leaked} is the mean effective n value based on Theorem 4.3 across all ratios excluding 0 and 1. Size for ogbn-arxiv refers to number of nodes, and the average number of nodes for Chord.

joint data splits between $\mathcal{A}$ and $\mathcal{T}$, helps ensure that any distinguishing power we observe is actually distribution inference, rather than inadvertent dataset inference.

Our experimental datasets were selected to incorporate common benchmarks (Census, CelebA) to enable comparisons with previous work, new datasets to assess more realistic property inference threats (RSNA Bone Age), as well as applying our definitions beyond ratio-based properties on graphs (mean node-degree on ogbn-arxiv, clustering coefficient on Chord). As noted by Zhang et al. [9], the target properties for a distribution inference attack can be either related to or independent of the task, and can be either explicit or latent features of the input data. For instance, attributes varied for Census are feature-based properties since these attributes are directly used as features for the models trained on them. On the other hand, an attribute like the age of a person is unrelated to detecting smiles and is a latent property that is not directly encoded as an input feature in the training data (but is available for our CelebA experiments from provided metadata).

Census [24] consists of several categorical and numerical attributes like age, race, education level to predict whether an individual's annual income exceeds \$50K. We focus on the ratios of whites (race) and females (sex) as properties and use three-layer feed-forward networks.

RSNA Bone Age [25] contains X-Ray images of hands, with the task being predicting the patient's age in months. We convert the task to binary classification based on an age threshold and use a pre-trained DenseNet [26] model for feature extraction, followed by a two-layer network for classification. We focus on the ratios of the females (available as metadata, but implicit in the examples) as properties.

CelebA [18] contains face images of celebrities, with multiple images per person. Each image is annotated with forty attributes such as gender, sunglasses, and facial hair. We use two different tasks, smile detection and gender prediction, and train convolutional neural networks from scratch for this dataset. We focus on the ratios of the females (smile-detection task) and old people (gender-prediction task) as properties. These attributes are provided as meta-data in the dataset. We create a network architecture with five convolutional layers and pooling layers followed by three linear layers, which is the smallest one we could find with reasonable task accuracy.

The **ogbn-arxiv** [27] dataset is a directed graph, representing citations between computer science arXiv papers. The task is to predict the subject area categories. We infer the mean node-degree property of the graph using four-layer Graph Convolutional Networks [28].

Chord [29] contains botnets with the Chord [30] topology artificially overlaid on top of background network traffic from CAIDA [31]. The dataset contains multiple graphs, with the task of detecting bot nodes in the graphs. We focus on inferring whether the underlying graphs (onto which we overlay botnets) have average clustering coefficients within a specific range. Following the model architecture proposed in Zhou et al. [29], we implement a Graph Convolutional architecture.

6.2 Attack Details

We perform each experiment ten times and report mean values with standard deviation in all of our experiments. Since the Loss Test uses a fixed test set per experiment, its results show no variation. For each dataset, we train 1000 victim models per distribution.

Loss Test. The adversary uses its test data to sample the two test sets S_0 and S_1. Since we use the same test data in evaluations, we turn off sampling while generating data with desired properties for this setting.

Threshold Loss. The adversary trains 50 models per distribution on its data split.

Meta-Classifier. We used Permutation Invariant Networks as our meta-classifier architecture [12]. The simulated adversary produces 800 models per distribution using its split of data to train the meta-classifier. For the case of CelebA, we use our extension of the Permutation Invariant Network that is compatible with convolutional layers (Section 5.2.2). Following experimental designs from prior works, we were able to achieve the accuracies that the authors reported (Section 6.3.1). However, using our experimental design leads to significantly lower distinguishing performance. Steps like ensuring no overlap in victim/adversary data, randomly sampled datasets for $\mathcal{G}_0(\mathcal{D})$ and $\mathcal{G}_1(\mathcal{D})$, and ensuring the same dataset size are necessary to avoid the risk that the meta-classifier is identifying something different about the distributions other than the claimed property. We think these steps are important for realistic experiments, so report the distinguishing accuracies based on this experimental design, even if they are lower for the same attacks than the results reported for the same tasks in previous work.

6.3 Binary Properties

We fix $\mathcal{G}_0$ and try the attacks on a range of $\mathcal{G}_1$ distributions for the first set of experiments. In Section 6.3.2, we report on experiments varying both distributions. Most prior works on distribution inference use arbitrary ratios, like distinguishing between 42% and 59% males [12]. Only recently have works started transitioning to more controlled experimental settings, like comparable dataset sizes for the victim and adversary and non-overlapping data sampling [6]. While having one of the ratios corresponding to the estimate of the underlying data distribution is justified, fixing the other arbitrarily makes it hard to understand the adversary's capabilities—we want to understand how dissimilar the distributions must be in order to be distinguishable. Additionally, the lack of keeping ratios consistent in experiments across properties makes it harder to compare information leakage across properties. Analyzing such trends is important for understanding how much of these properties are leaked across different configurations (explicit attribute, latent property) and assess the adversary's capabilities under different scenarios (black-box access, white-box access).

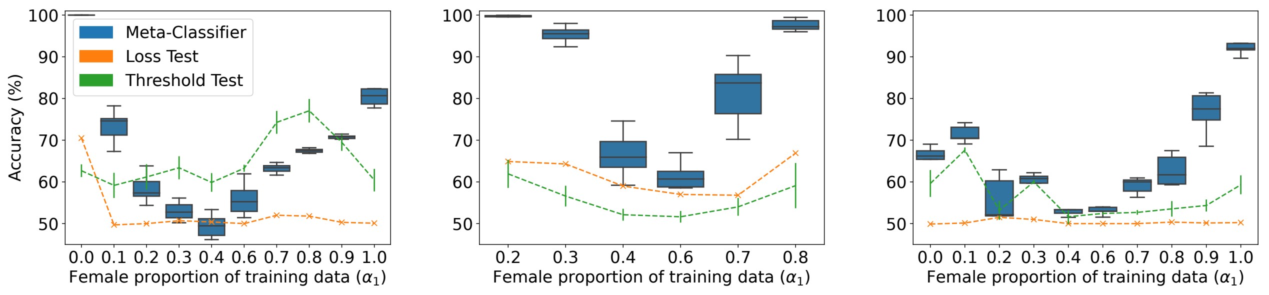
Experiments with binary classification for properties can provide useful insights into the effectiveness of distribution inference attacks, but most realistic attacks would not be based on distinguishing between two known distributions. In Section 6.4, we consider attacks that can infer the underlying ratio without any prior assumptions about distinguishing particular distributions.

6.3.1 Distinguishing Imbalanced Ratios

Since the original ratios for the targeted property may be unbalanced in the dataset (Section 3), for these experiments we fix $\mathcal{G}_0$ to a balanced ($\alpha_0 = 0.5$) ratio for the chosen attribute for the Census, CelebA, and RSNA Bone Age datasets. Then, we vary α_1 to evaluate inference risks and understand how well an adversary could distinguish between models trained using distributions with different proportions of the targeted property.

Census. We summarize the accuracies for the three attack methods across varying proportions of females in Figure 2a and whites in Figure 10a. The Loss Test performs only marginally better than random guessing in most cases for females, yielding n_{leaked} values in the range $[0, 0.03]$, essentially close to random guessing. On the other hand, the Threshold Test and meta-classifiers are able to achieve non-trivial distinguishing accuracies for the female proportion, with n_{leaked} values in ranges $[0.1, 1.2]$ and $[0.02, 6.5]$ respectively, with a similar median n_{leaked} value of 0.33, showing how the two are not very far apart in effectiveness. Leakage increases as the distributions become more disparate, with near-perfect distinguishing accuracy for the extreme case of $\alpha_1 = 0$. For race, none of the attacks detect anything ($n_{\text{leaked}} \approx 0$) apart from the surprising results on Threshold Test, which performs asymmetrically well, approaching 80% accuracy for mostly-white distributions.

Comparison with previous results. Ganju et al. [12] applied their meta-classifier method on two properties on this dataset: 38% vs. 65% women (case A), and 0% vs.



(a) Census **(b)** RSNA Bone Age **(c)** CelebA

Fig. 2. Classification accuracy for distinguishing proportion of females in training data for (a) Census, (b) RSNA Bone Age, and (c) CelebA. The RSNA Bone Age dataset does not include ratios below 0.2 or above 0.8, since sampling the original dataset for that ratios produces datasets that are too small to train models with meaningful performance. Performance of the attacks increases as the distributions diverge (α_1 moves away from 0.5), but is not symmetric.

87% whites (case B). For case B, the Loss Test performs as well as random guessing (50.1%), while the Threshold Test ($92.4 \pm 2.6\%$) approaches meta-classifier performance ($99.9 \pm 0.1\%$), achieving a high (compared to $\alpha = 0.5$ experiments) $n_{\text{leaked}} \approx 11$. For case A, the Threshold Test ($62.7 \pm 2.0\%$) outperforms meta-classifiers ($62.1 \pm 1.7\%$) while achieving $n_{\text{leaked}} \approx 0.03$, barely better than random guessing , and the Loss Test method fails (50%). Ganju et al. report 97% accuracy for case A and 100% for case B. We were able to closely reproduce these results in their setting which includes overlapping data between victim and adversary and does not ensure changes in ratios do not affect dataset size or class imbalance. In the more realistic experimental design in which we ensure there is no victim/adversary overlap and maintain the label ratios and same dataset sizes (Section 6.2), the accuracies are much lower—for example, in case A the distinguishing accuracy is 97% using their experimental design but drops to 62% when more carefully prepared datasets are used in our design. These results suggest that although the distinguishing accuracy is high between the two distributions in the tests in their setting, the attacks are not actually inferring the intended property but are predicting the distribution based on other differences between the datasets.

RSNA Bone Age. Distinguishing accuracies for the female proportion on the RSNA Bone Age dataset are plotted in Figure 2b. The simple Loss Test performs nearly as well as the Threshold Test leaking a median of $n_{\text{leaked}} \approx 0.2$ and $n_{\text{leaked}} \approx 0.1$ respectively. The meta-classifier attack, on the other hand, has a much higher leakage of $n_{\text{leaked}} \approx 6$.

CelebA. Figure 2c shows the distinguishing accuracy for the CelebA data on proportion of females, and Figure 10b for the proportion of examples marked as "old". The Threshold Test performs much worse compared to Census and RSNA Bone Age, with the median $n_{\text{leaked}} < 0.05$, compared to ≈ 0.7 using meta-classifiers. Figure 11 shows meta-classifier prediction accuracy for three different representations of the shadow models used to train the meta-classifier: using parameters from only linear layers, only convolutional layers, and all layers of the models. While inferring the ratio of old people (Figure 11a), including just the fully-connected layers works best, yielding $n_{\text{leaked}} \approx 0.12$ and not too far off from using all layers or just the convolutional layers ($n_{\text{leaked}} \approx 0.07$). For the case of sex ratios (Figure 11b), using the full model helps extract more information ($n_{\text{leaked}} \approx 0.32$) than either of the convolutional ($n_{\text{leaked}} \approx 0.24$) or linear ($n_{\text{leaked}} \approx 0.05$) layers. These trends suggest the likelihood of some layers' parameters capturing specific property-related information better than the others. Linear layers are more helpful for ratios of old people whereas for ratios of females, convolutional layers are significantly better.

6.3.2 Varying Proportions

An adversary may not necessarily be interested in distinguishing between the balanced case ($\alpha = 0.5$) and other ratios. For instance, health datasets for specific ailments may have a higher underlying prevalence in females, and the adversary may be interested in differentiating between two particular ratios of females, like 0.3 and 0.4. We thus experiment with the case where both distributions are varied: $\mathcal{G}_0(\mathcal{D})$ and $\mathcal{G}_1(\mathcal{D})$ with corresponding ratios α_0 and α_1 respectively. Distribution inference risk seems particularly acute when an adversary can distinguish the proportion of an uncommon property. Observing performance trends as the difference in ratios increases also helps us understand how much of a threat property inference may pose as the similarity of the distributions varies.

Figure 12 shows the distinguishing accuracies (and corresponding n_{leaked} values) between models (in the form of heatmaps) trained on distributions $\mathcal{G}_0(\mathcal{D})$ and $\mathcal{G}_1(\mathcal{D})$ while varying corresponding α_0 (horizontal axis) and α_1 (vertical axis), for ratios of females on Censusand RSNA Bone Age. For instance, in Figure 12a, $(\alpha_0, \alpha_1) = (0.2, 0.9)$ in the upper-right triangle correspond to meta-classifier performance (70%), while $(\alpha_0, \alpha_1) = (0.9, 0.2)$ in the lower-left triangle gives the corresponding $n_{\text{leaked}} = 0.17$. Entries along a given diagonal have the same value of $|\alpha_0 - \alpha_1|$. As reflected in the heatmap colors, distinguishing accuracies are roughly the same along diagonals. The variance in performance across runs is relatively high for similar distributions (small $|\alpha_0 - \alpha_1|$) and decreases as the distributions diverge. For CelebA (sex) and Census (sex), we observe that $n_{\text{leaked}} < 1$ in most cases. These small values do not imply the inability of *any* adversary to distinguish between the distributions, only that for the given attacks we observe little information leakage.

6.4 Direct Regression over α

Inspired by Zhou *et al.* [6], we performed a direct regression experiment in which we trained the meta-classifiers to predict α directly. This corresponds to a more realistic attack setting for many scenarios than one in which the adversary is distinguishing between two predefined α values. For this experiment, we construct a training dataset for the regression meta-classifier with tuples of the form (M_α, α), where M_α is some model with a training distribution corresponding to the ratio α. We train the meta-classifier using M_α models for all the ratios α that we experiment with in Section 6.3.2 ($\{0.0, 0.1, \ldots, 1.0\}$ for Census and CelebA, and $\{0.2, \ldots, 0.8\}$ for RSNA Bone Age). The meta-classifier follows the same permutation-invariant architecture as in the binary property experiments (Section 5.2), just with a mean squared error (MSE) loss for training.

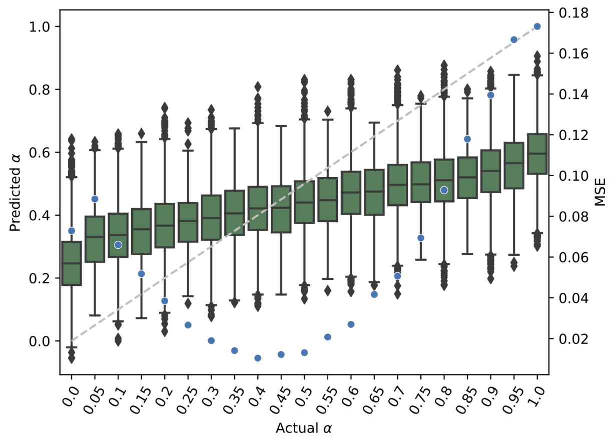
Figure 3 shows the distribution of the predictions of the regression meta-classifiers and Table 2 reports the MSE and n_{leaked} values. For all these experiments, we train meta-classifiers five times with different seeds, and report aggregate results over all the meta-classifiers and victim models, for each dataset and property. In the plots in Figure 3, we include results for actual α values at both tenths and the intermediate 0.05 ratios to confirm that the meta-classifiers are indeed learning to predict α and not just overfitting to the α values observed in training. The meta-classifiers do exhibit a bias toward balanced predictions, showing a smooth curve for the MSE values with a minimum near $\alpha = 0.5$.

Dataset	Attribute	n_{leaked} (B)	(BR)	(R)	MSE
Census	Sex	0.2	0.3	8.8	0.053
	Race	0.1	0.2	6.0	0.091
CelebA	Sex	0.3	0.4	10.6	0.030
	Age	0.2	0.4	5.1	0.047
RSNA Bone Age	Sex	6.2	15	269.4	0.001

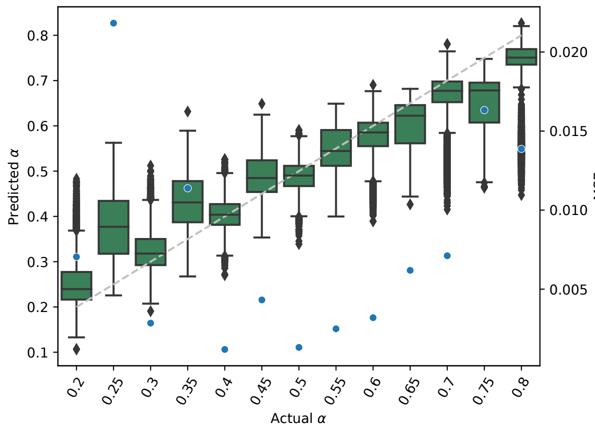
Table 2. Median n_{leaked} values when using binary meta-classifiers n_{leaked} (B), regression meta-classifiers n_{leaked} (R), and regression meta-classifiers for binary predictions n_{leaked} (BR), along with average Mean Squared Error (MSE) for direct regression over α. n_{leaked} is nearly double for all cases that use regression meta-classifiers for binary predictions, when compared to the binary meta-classifiers.

The attacks are quite successful in most cases, achieving $n_{\text{leaked}} > 5$ for all of our settings, and surprisingly high leakage for RSNA Bone Agewith $n_{\text{leaked}} > 260$. These regression attacks show that adversaries can infer sensitive information about training datasets even in the more realistic setting where adversaries do not have prior knowledge of distributions. However, the attacks are not always successful— performance for meta-classifiers on Census (race) is not much better (Figure 3a) than that when guessing $\alpha = 0.5$ blindly (expected MSE 0.1).

Given the high n_{leaked} values for the regression tests, we tried using the regression meta-classifiers to distinguish between binary properties. To produce a classification between models with training distribution ratios α_0 and α_1, a regression meta-classifier $M_{\text{regression}}$'s prediction for some model m is converted to a binary outcome by simply checking which of the two considered distribution ratios the predicted ratio is closer to: $\hat{b} = \mathbb{I}\left[M_{\text{regression}}(m) \geq \frac{\alpha_0+\alpha_1}{2}\right]$. Each entry in Table 2 is averaged over 5 trials × 100 victim models × 11 ratios ($\{0.0, 0.1, \ldots, 0.9, 1.0\}$ for Census; 7 in the case of RSNA Bone Age). In most cases, the accuracy improves significantly over the binary classifiers, with the n_{leaked} value nearly doubling for most settings. For instance, the clas-



(a) Census (Sex)



(b) RSNA Bone Age

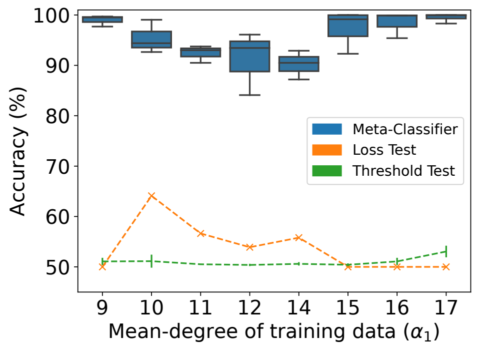
Fig. 3. Predicted α values (left y-axis) for models with training distributions for varying α values (x-axis), for all victim models and regression meta-classifier experiments (green box-plots), along with mean squared error (right y-axis labels, with different scales on the two graphs, and blue dots), for (a) Census and (b) RSNA Bone Age datasets. with The diagonal gray dashed line represents the ideal case, where the regression classifier perfectly predicts α. For each ratio of the form $0.05 \cdot x$ (for varying x), we train regression meta-classifiers 5 times with different seeds, and test 100 victim models. As indicated by n_{leaked} values, the RSNA Bone Age dataset observes very good performance, with nearly all predictions lining up with the diagonal, while for Census (sex), predicted ratios are usually in $[0.2, 0.6]$.

sification accuracy increases by $\sim$ 4% and $\sim$ 15% for CelebA (sex) and RSNA Bone Age respectively, corresponding to an increase of n_{leaked} by ~ 0.14 for CelebA (sex) and $\sim$ 8.51 for RSNA Bone Age. This improvement is not surprising since the binary attack uses models only from two distributions, whereas the regression attack has models from a wide range of alpha values and thus can learn more. Figure 13 shows the accuracies and n_{leaked} values for using specific ratio binary classifiers compared with the improved accuracies obtained using the regression meta-classifier. The n_{leaked} values for each pair of ratios (α_0, α_1) shows how these improvements are uniform across all pairs of distributions.

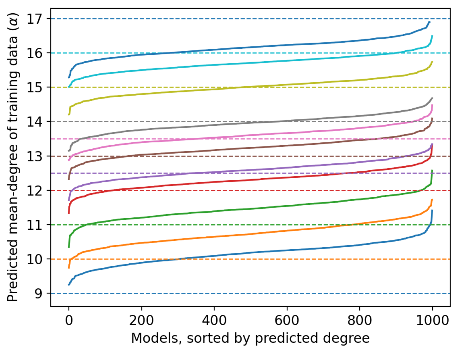
6.5 Graph Properties

Our experiments using both binary and regression classifiers on the graph datasets reveal surprisingly high property leakage. Observed n_{leaked} values for ogbn-arxiv are much higher (100-200 range) than was observed in the experiments on tabular and image datasets (with the exception of RSNA Bone Age). (For the clustering coefficients on Chord, we do not have a way to compute n_{leaked}, but also see evidence of substantial leakage.)

ogbn-arxiv. For the ogbn-arxiv dataset, we set $\mathcal{G}_0$ such that the graph has a mean node-degree of $\alpha_0 = 13$, and for $\mathcal{G}_1$, modify the graph to have a mean-degree α_1 as an integer in the range [9, 17]. We produce test datasets by pruning either high or low-degree nodes from the original graph to achieve a desired α_1. Like the other datasets, meta-classifier performance increases as the distributions diverge, albeit with much smaller drops. Both the Loss Test and Threshold Test fail on this dataset with n_{leaked} values below 1, compared to the meta-classifiers (Figure 4a) which leaks $n_{\text{leaked}} \approx 40$ in most cases. The attacks leak much more information as the degrees increase than when they decrease— n_{leaked} values are nearly double for (12, 14) than (12, 13) as the two mean node-degrees, despite having comparable distinguishing accuracies. Motivated by the success of regression attacks for ratio-based properties (Section 6.4), we also trained a regression variant of the meta-classifier to predict the average degree of the training graph directly. The resulting meta-classifier performs quite well (Figure 4b), achieving a mean squared error (MSE) loss of 0.393±0.36. It generalizes well to unseen distributions, achieving an average MSE loss of 0.076 for $\alpha = 12.5$ and 13.5. A property inference adversary can thus be strong enough to directly predict the average node degree of the training distribution. Similar to the case of ratios, we try different combinations of mean node-degree values by setting different mean node-degrees α_0 and α_1 (Figure 5). As apparent, n_{leaked} has a wide spread in its values across different distributions—starting from ≈ 3 to approaching infinity, with a median of ≈ 31.

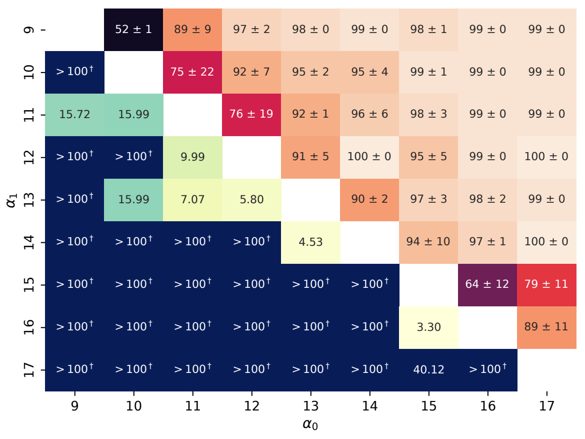


(a) Distinguishing accuracy ($\alpha_0 = 13$) as mean degree varies.



(b) Mean node-degree α_1 as predicted by the meta-classifier.

Fig. 4. Performance for (a) distinguishing between models with different mean node-degrees in training data, and (b) directly inferring the mean node-degree of the training data, for the ogbn-arxiv dataset. Each color represents the true degree (dashed lines) of the models being tested. The meta-classifier attack is remarkably successful on this dataset and further accentuates how some attacks can infer underlying properties nearly exactly on some datasets.



α_1 \ α_0	9	10	11	12	13	14	15	16	17
9		52 ± 1	89 ± 9	97 ± 2	98 ± 0	99 ± 0	98 ± 1	99 ± 0	99 ± 0
10	> 100†		75 ± 22	92 ± 7	95 ± 2	95 ± 4	99 ± 1	99 ± 0	99 ± 0
11	15.72	15.99		76 ± 19	92 ± 1	96 ± 6	98 ± 3	99 ± 0	99 ± 0
12	> 100†	> 100†	9.99		91 ± 5	100 ± 0	95 ± 5	99 ± 0	100 ± 0
13	> 100†	15.99	7.07	5.80		90 ± 2	97 ± 3	98 ± 2	99 ± 0
14	> 100†	> 100†	> 100†	> 100†	4.53		94 ± 10	97 ± 1	100 ± 0
15	> 100†	> 100†	> 100†	> 100†	> 100†	> 100†		64 ± 12	79 ± 11
16	> 100†	> 100†	> 100†	> 100†	> 100†	> 100†	3.30		89 ± 11
17	> 100†	> 100†	> 100†	> 100†	> 100†	> 100†	40.12	> 100†	

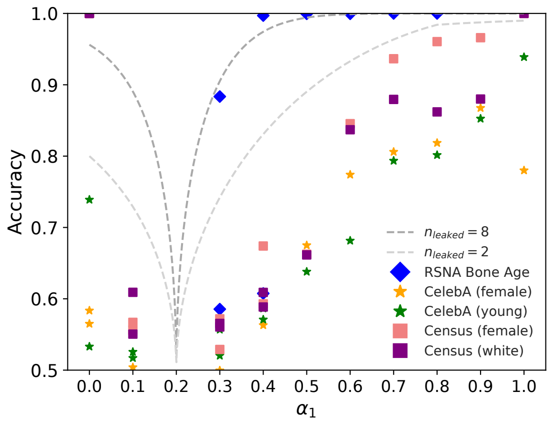
Fig. 5. Effectiveness of meta-classifiers on ogbn-arxiv dataset. n_{leaked} values (bottom-left triangles) and mean node-degree (degrees α_0, α_1) in training data.

Chord. For the Chord dataset, we construct $\mathcal{G}_0$ to have graphs with average clustering-coefficient < 0.0061 and $\mathcal{G}_1$ with average clustering-coefficient above > 0.0071. We pick these values to minimize the overlap between the two distributions, while maintaining a decent accuracy on the original task. For the case where both the trainer's and adversary's datasets are sampled from the same pool of data, the adversary has near-perfect distinguishing accuracy, even when training the meta-classifier with ten models (and testing on 1000). However, the adversary cannot achieve the same level of performance in the absence of data overlap. Using the Loss Test yields an accuracy of 63%, while the Threshold Test and meta-classifier struggle to perform better than random (≈51%). This disparity in performance further justifies our experimental design choice to consider non-overlapping data splits—evaluating property inference attacks on models trained from the same dataset pool seems effective, but it cannot distinguish learning some unrelated property from the claimed inference.

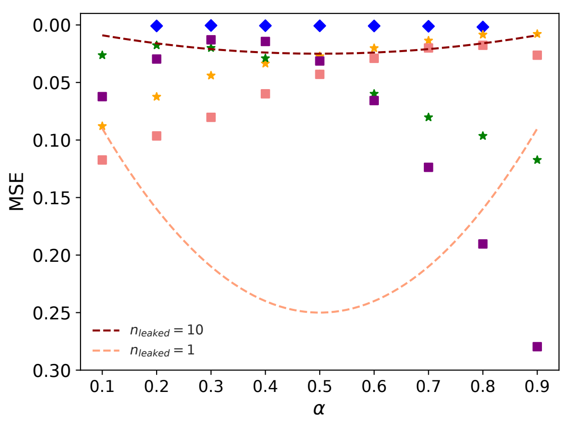
6.6 Summary of Experimental Results

Figure 6 summarizes the distribution leakage observed for our ratio and regression based experiments. Most attacks across varying distributions and datasets correspond to values of $n_{\text{leaked}} \approx 1$ for classification, and n_{leaked} between 5 and 10 for regression. The low n_{leaked} values for binary classifiers show how little information those attacks are able to leak from the model, less than the equivalent of sampling just two records from the training distribution in all cases except RSNA Bone Age, but the regression results do indicate that even in these settings non-trivial information is leaking and there are opportunities for better inference attacks. The biggest exception is for RSNA Bone Age, where we observe n_{leaked} values above 10 for the binary classifiers (Figure 12b) and up to 270 for the regression meta-classifier, and the graph datasets, where they are in the hundreds for ogbn-arxiv (Figure 5).

Dataset Inference Susceptibility. This leakage is much higher for RSNA Bone Age than it is for similar Boolean ratio-like properties for datasets like Census and CelebA, suggesting this particular dataset is prone (much more than the others we explore) to high-confidence inference attacks, at least for the sex property. We do not yet have a good understanding of why this dataset leaks so much more distribution informa-



(a) Distinguishing accuracy for inference for $\alpha_0 = 0.2$ with varying α_1.



(b) Mean Square Error (MSE) for regression inference.

Fig. 6. Distinguishing accuracy (a) and MSE (b) for inference attacks with varying α_1 on the horizontal axis. The plotted results are for the most effective attacks from the experiments described in Section 6. The curves in (a) show the comparable distinguishing accuracy for $n_{\text{leaked}} = 2$ (indicating that most of the attacks are comparable to leaking fewer than two samples from the training distribution) and $n_{\text{leaked}} = 8$, showing that a few of the attacks on the RSNA Bone Age dataset (and extreme attacks on Census for the race attribute) do leak a substantial amount of information. Similar trends hold for regression, with n_{leaked} somewhere between 1 and 10 for most cases (b). The highest leakages we observed are for the graph datasets, not shown in this figure.

tion than the other two, but think the causes of variation in inference risk is an important question for future study. Although we observe an increase in n_{leaked} values across all our experiments as the distributions diverge from each other, these trends are not symmetric. Hence, not only do more divergent distributions seem to leak more information, that information is more valuable to an adversary since as the distributions diverge less information is needed to distinguish them accurately. This might be suggestive of biased learning algorithms, or perhaps the attacks' capabilities to infer distributional properties effectively.

Attack Comparison. Analyzing n_{leaked} values for different attacks (one of the primary purposes of the notion) shows there is no clear winner out of the three attacks. n_{leaked} serves as a measure for comparing the power of different attacks across ranges of distributions—seemingly different distinguishing accuracies can correspond to very close n_{leaked} values (for instance, CelebA), while similar ones can correspond to very different n_{leaked} values (for instance, ogbn-arxiv). The lack of any clear ranking of the simple black-box and white-box attacks shows why it is necessary to include simple baselines when evaluating property inference risk. In fact, the inexpensive Threshold Test does better than meta-classifiers for about 30% of our experiments across all the datasets. There are many instances where either of the two simple black-box tests perform exceptionally well—even better than meta-classifier. For instance, Threshold Test on Census (race) and Loss Test on CelebA (old) outperform the meta-classifiers for nearly all of pairs of ratios. These observations further support the approach of using more direct attacks like Loss Test and Threshold Test before training expensive meta-classifiers, and also considering attacks that combine meta-classifiers with black-box attacks.

Peculiar Trends. We also observe some trends specific to a dataset and property, but can only speculate on their causes. For example, on the Census dataset, the adversary has notably high accuracy in differentiating between distributions when one is without any females ($\alpha_0 = 0$) or males ($\alpha_0 = 1$) with distinguishing accuracies close to 100%, regardless of the actual proportion of females in the data. Detecting the mere presence or absence of members with a particular attribute is much easier than trying to deduce the exact ratio of members with that attribute, which is unsurprising for attributes that impact the task predictions. Similarly, a difference in ratios of ≥ 0.3 on RSNA Bone Age (Figure 12b) yields $> 90\%$ accuracy for all cases using meta-classifiers, with n_{leaked} values ≥ 7, going up to perfect distinguishing accuracy. Unlike Census, performance on CelebA at the extremes (no males or females when inferring sex ratios, and no young or old people when inferring old ratios) is far from perfect. This may be because features like race and gender in Census are directly used for model training, and thus their presence or absence would directly impact both predictions and model parameters. Whereas for CelebA, the complicated feature extractor may not explicitly capture these latent (and inherently ambiguous) properties.

Regression for Binary Classification. Comparing n_{leaked} values for the binary meta-classifiers and regression-based meta-classifiers tuned for binary classification demonstrates how additional information about the underlying ratios can have a huge impact on leakage. Having models trained on training distributions for a wide range of α can help ensure the meta-classifier actually learns to infer the underlying ratios, compared to the binary classification case where it is most likely to rely on specific signals just to distinguish between two given distributions. Although that is indeed the given task, the ability to capture the association between α and the desired predictions can, and does, help the meta-classifiers improve their performance. Note that training the regression-based meta-classifiers does not require a stronger threat model than is assumed for the binary classifier case. In both cases, the adversary needs access to a training distribution with enough samples to be able to create representative datasets for different distributions. Training the regression meta-classifier requires more computational resources (training models for multiple ratios) than is required to train the binary meta-classifier (training models only for two ratios), but does not otherwise require a stronger adversary.

7 Conclusions

A important step to developing understanding of distribution inference risks is a precise and formal definition, and we found the general definition we introduce to be useful for conceptualizing the space of attacks especially in having a precise way to separate intended statistical inference from distribution inference attacks. The definition also leads to a systematic approach to quantifying the leakage from distribution inference attacks. Our empirical results reveal how intuition may not necessarily align with actual observations. Seemingly similar pairs of distributions can have starkly different attack success rates, and simple attacks with limited access can sometimes outperform computationally expensive meta-classifiers. Our experiments also show how direct regression of underlying ratios of training distributions is a real threat, and can be used to improve the performance of binary distinguishing attacks.

Our work raises more questions than it answers—*why do some models leak a lot of information about certain properties of their training distribution but others leak little*, *what are the limits on how precisely training distributions can be distinguished*, *why do models trained on some datasets (like RSNA Bone Age and the graphs) appear to leak so much more information than others*. We are not able to answer these questions yet, although our experiments provide several intriguing observations and suggest possibilities to explore. It is not surprising that so little is understood about distribution inference—the research community has put extensive effort into studying membership inference attacks, and we are just beginning to be able to understand how and why membership inference risk varies [32]. There could also be trade-offs between robustness, fairness, interpretability, and vulnerability to distribution inference attacks. For instance, a model that is fair with regards to some attribute may be less vulnerable to distribution inference attacks on that attribute.

Another aspect of these attacks is robustness to different training processes. It is unclear how factors like overfitting, training for robustness, or data augmentation can impact property inference risk. Exploring how these factors increase or decrease susceptibility is part of ongoing work and may pave the way for understanding these attacks better, perhaps even leading to principled and effective defenses.

We expect there is room for improving distribution inference attacks, and hope our results raise awareness that distribution inference attacks that expose sensitive aspects of training data are possible and require further exploration, analysis, and development of mitigations.

Acknowledgements

The authors thank Yifu Lu and Justin Chen for contributing to some of the experiments in this paper, along with Xiao Zhang, Bargav Jayaraman, and Ambar Pal for helpful feedback on the paper including a detailed reading of our proof. This work was partially supported by grants from the National Science Foundation (#1717950 and #1915813), and funding from Lockheed Martin.

Availability

Code for reproducing our experiments is available at: https://github.com/iamgroot42/FormEstDistRisks.

References

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," in *IEEE Symposium on Security and Privacy*, 2017.

[2] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[3] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," in *USENIX Security Symposium*, 2014.

[4] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[5] M. Chase, E. Ghosh, and S. Mahloujifar, "Property Inference from Poisoning," *arXiv:2101.11073*, 2021.

[6] J. Zhou, Y. Chen, C. Shen, and Y. Zhang, "Property Inference Attacks Against GANs," *arXiv:2111.07608*, 2021.

[7] D. Gopinath, H. Converse, C. Pasareanu, and A. Taly, "Property Inference for Deep Neural Networks," in *IEEE/ACM International Conference on Automated Software Engineering*, 2019.

[8] M. Jegorova, C. Kaul, C. Mayor, A. Q. O'Neil, A. Weir, R. Murray-Smith, and S. A. Tsaftaris, "Survey: Leakage and Privacy at Inference Time," *arXiv preprint arXiv:2107.01614*, 2021.

[9] W. Zhang, S. Tople, and O. Ohrimenko, "Leakage of Dataset Properties in Multi-Party Machine Learning," in *USENIX Security Symposium*, 2021.

[10] P. Maini, M. Yaghini, and N. Papernot, "Dataset Inference: Ownership Resolution in Machine Learning," in *International Conference on Learning Representations*, 2021.

[11] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting," in *IEEE Computer Security Foundations Symposium*, 2018.

[12] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations," in *ACM Conference on Computer and Communications Security*, 2018.

[13] D. Desfontaines and B. Pejó, "SoK: Differential privacies," in *Privacy Enhancing Technologies Symposium*, 2020.

[14] D. Kifer and A. Machanavajjhala, "Pufferfish: A Framework for Mathematical Privacy Definitions," *ACM Transactions on Database Systems (TODS)*, 2014.

[15] W. Zhang, O. Ohrimenko, and R. Cummings, "Attribute Privacy: Framework and Mechanisms," *arXiv:2009.04013*, 2020.

[16] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate Estimation of the Degree Distribution of Private Networks," in *IEEE International Conference on Data Mining*, 2009.

[17] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI Trojans Using Meta Neural Analysis," in *IEEE Symposium on Security and Privacy*, 2021.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale CelebFaces Attributes (CelebA) Dataset," 2018.

[19] D. Pasquini, G. Ateniese, and M. Bernaschi, "Unleashing the Tiger: Inference Attacks on Split Learning," in *ACM Conference on Computer and Communications Security*, 2021.

[20] Z. Zhang, M. Chen, M. Backes, Y. Shen, and Y. Zhang, "Inference Attacks Against Graph Neural Networks," in *USENIX Security Symposium*, 2022.

[21] L. Lyu and C. Chen, "A Novel Attribute Reconstruction Attack in Federated Learning," *arXiv:1712.00409*, 2021.

[22] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, 2020.

[23] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep Sets," *Advances in Neural Information Processing Systems*, 2017.

[24] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 81–85, 2000.

[25] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, A. B. Mamonov, A. Bilbily, M. Cicero, I. Pan, L. A. Pereira, R. T. Sousa, N. Abdala *et al.*, "The RSNA Pediatric Bone Age Machine Learning Challenge," *Radiology*, vol. 290, no. 2, pp. 498–503, 2019.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[27] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft Academic Graph: When experts are not enough," *Quantitative Science Studies*, vol. 1, no. 1, pp. 396–413, 2020.

[28] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *International Conference on Learning Representations*, 2017.

[29] J. Zhou, A. M. Xu, Zhiying amd Rush, and M. Yu, "Automating Botnet Detection with Graph Neural Networks," *AutoML for Networking and Systems Workshop of MLSys 2020 Conference*, 2020.

[30] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[31] Center for Applied Internet Data Analysis, "The CAIDA UCSD Anonymized Internet Traces," https://www.caida.org/data/passive/passive_dataset.xml, 2018.

[32] B. Kulynych, M. Yaghini, G. Cherubin, M. Veale, and C. Troncoso, "Disparate Vulnerability to Membership Inference Attacks," in *Privacy-Enhancing Technologies Symposium*, 2022.

[33] Mathematics State Exchange User 'user13888', "What is the relationship of $\mathcal{L}_1$ (total variation) distance to hypothesis testing?" Mathematics Stack Exchange, https://math.stackexchange.com/q/72730, 2011.

[34] H. Jain, "Applications of Pinsker's inequality," Harvard Course Notes, http://people.seas.harvard.edu/~madhusudan/courses/Spring2016/scribe/lect07.pdf.

[35] A. B. Tsybakov, "Introduction to Nonparametric Estimation," 2009.

[36] V. Papyan, X. Y. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, 2020.

A Proofs

A.1 Proof of Lemma 4.1

Assume the adversary can fully recover a dataset S (of size n) from some model M trained on it. Assume $\psi(\cdot)$ is an estimator for testing the hypothesis, i.e. $\psi(S) = b_{\in\{0,1\}}$ means that S comes from $\mathcal{G}_b(\mathcal{D})$. Assuming an equal likelihood of the chosen dataset S being from either distributions, we have:

$$\begin{aligned}\text{Error} =&\frac{1}{2}\big(\Pr_{S\leftarrow\mathcal{G}_0(\mathcal{D})^n}\big[\psi(S)=1\big]+\\ &\Pr_{S\leftarrow\mathcal{G}_1(\mathcal{D})^n}\big[\psi(S)=0\big]\big) \qquad (9)\\ =&\frac{1}{2}(\text{Type I Error} + \text{Type II Error}).\end{aligned}$$

Combining with the result from [33]:

$$\text{Error} \geq \frac{1}{2} - \frac{1}{2}\delta(\mathcal{G}_0(\mathcal{D})^n, \mathcal{G}_1(\mathcal{D})^n) \qquad (10)$$

$$\Rightarrow \text{Accuracy} \leq \frac{1}{2} + \frac{1}{2}\delta(\mathcal{G}_0(\mathcal{D})^n, \mathcal{G}_1(\mathcal{D})^n), \qquad (11)$$

where $\delta()$ is the total variation distance between two probability measures, and $\mathcal{G}_{b\in\{0,1\}}(\mathcal{D})^n$ refers to the distribution of n samples from $\mathcal{G}_{b\in\{0,1\}}(\mathcal{D})$. Thus, the maximum accuracy while differentiating between datasets sampled from either distribution is bounded by the total variation distance between them. Let $\rho_b(x)$ be the generative probability density function for some sample x drawn from $\mathcal{G}_b(\mathcal{D})$, for $b \in \{0,1\}$. This density function can then be broken down into a multinomial distribution and priors as:

$$\rho_b(x) = (1-\alpha_b)p_b(\mathbf{x}|0) + \alpha_b p_b(\mathbf{x}|1), \qquad (12)$$

where α_b is the prior for $p(1)$ corresponding to $\mathcal{G}_b(\mathcal{D})$, and $\rho_b(x|1)$ is the associated conditional generative probability density function. Note that $p_0(\mathbf{x}|0) = p_1(\mathbf{x}|0)$ and $p_0(\mathbf{x}|1) = p_1(\mathbf{x}|1)$, since they both come from the underlying distribution $\mathcal{D}$. Without loss of generality, let $\alpha_0 > \alpha_1$ (we omit the case of same ratios, since that is trivially indistinguishable). Then:

$$\begin{aligned}&\alpha_1\rho_0(x) - \alpha_0\rho_1(x) && (13)\\ &= \alpha_1((1-\alpha_0)p_0(\mathbf{x}|0) + \alpha_0 p_0(\mathbf{x}|1)) && (14)\\ &\ - \alpha_0((1-\alpha_1)p_1(\mathbf{x}|0) + \alpha_1 p_1(\mathbf{x}|1)) &&\\ &= \alpha_1 p_0(\mathbf{x}|0) - \alpha_1\alpha_0 p_0(\mathbf{x}|0) + \alpha_0\alpha_1 p_0(\mathbf{x}|1) && (15)\\ &\ - (\alpha_0 p_1(\mathbf{x}|0) - \alpha_0\alpha_1 p_1(\mathbf{x}|0) + \alpha_1\alpha_0 p_1(\mathbf{x}|1)) &&\\ &= (\alpha_1 - \alpha_0)p_0(\mathbf{x}|0) \leq 0 &&\\ &\Rightarrow \frac{\rho_0(x)}{\rho_1(x)} \leq \frac{\alpha_0}{\alpha_1} && (16)\end{aligned}$$

Using this inequality, the relative entropy (KL divergence) from $\mathcal{G}_1(\mathcal{D})$ to $\mathcal{G}_0(\mathcal{D})$ can be written as:

$$\begin{aligned}D_{KL}(\mathcal{G}_0(\mathcal{D}) \parallel \mathcal{G}_1(\mathcal{D})) &= \int \rho_0(x)\log\left(\frac{\rho_0(x)}{\rho_1(x)}\right)dx && (17)\\ &\leq \int \rho_0(x)\log\left(\frac{\alpha_0}{\alpha_1}\right)dx && (18)\\ &= \log\left(\frac{\alpha_0}{\alpha_1}\right)\int \rho_0(x)dx && (19)\\ &= \log\left(\frac{\alpha_0}{\alpha_1}\right) && (20)\end{aligned}$$

Since the function f is binary, a prior of α_b for $p(f(x) = 1)$ implies a prior of $(1-\alpha_b)$ for $p(f(x) = 0)$. Utilizing this symmetry, we can similarly upper-bound $D_{KL}(\mathcal{G}_1(\mathcal{D}) \parallel \mathcal{G}_0(\mathcal{D}))$ with $\log\left(\frac{1-\alpha_1}{1-\alpha_0}\right)$. Removing the $\alpha_0 \geq \alpha_1$ assumption and replacing with the max/min of these two appropriately, we get:

$$\begin{aligned}D_{KL}(\mathcal{G}_0(\mathcal{D}) \parallel \mathcal{G}_1(\mathcal{D})) &\leq \log\left(\frac{\max(\alpha_0,\alpha_1)}{\min(\alpha_0,\alpha_1)}\right) \qquad (21)\\ D_{KL}(\mathcal{G}_1(\mathcal{D}) \parallel \mathcal{G}_0(\mathcal{D})) &\leq \log\left(\frac{1-\min(\alpha_0,\alpha_1)}{1-\max(\alpha_0,\alpha_1)}\right)\end{aligned}$$

From [34], we know that:

$$D_{KL}(\mathcal{G}_0(D)^n \parallel \mathcal{G}_1(D)^n) = nD_{KL}(\mathcal{G}_0(D) \parallel \mathcal{G}_1(D)) \qquad (22)$$

Thus, when using a dataset S of size $|S| = n$, the equivalent KL-divergence can be bounded by:

$$\begin{aligned}D_{KL}(\mathcal{G}_0(\mathcal{D})^n \| \mathcal{G}_1(\mathcal{D})^n) &\leq n\log\left(\frac{\max(\alpha_0,\alpha_1)}{\min(\alpha_0,\alpha_1)}\right) \qquad (23)\\ D_{KL}(\mathcal{G}_1(\mathcal{D})^n \| \mathcal{G}_0(\mathcal{D})^n) &\leq n\log\left(\frac{1-\min(\alpha_0,\alpha_1)}{1-\max(\alpha_0,\alpha_1)}\right)\end{aligned}$$

Using the relation between total variation distance and KL-divergence [35], we know:

$$\delta(\mathcal{G}_0(\mathcal{D})^n, \mathcal{G}_1(\mathcal{D})^n) \leq \sqrt{1 - e^{-D_{KL}(\mathcal{G}_0(\mathcal{D})^n \; \| \; \mathcal{G}_1(\mathcal{D})^n)}} \tag{24}$$

$$= \sqrt{1 - e^{-nlog\left(\frac{\max(\alpha_0,\alpha_1)}{\min(\alpha_0,\alpha_1)}\right)}} \tag{25}$$

$$= \sqrt{1 - \left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)^n} \tag{26}$$

Similarly, using $D_{KL}(\mathcal{G}_1(\mathcal{D}) \;\|\; \mathcal{G}_0(\mathcal{D}))$ in the inequality above we get:

$$\delta(\mathcal{G}_0(\mathcal{D})^n, \mathcal{G}_1(\mathcal{D})^n) \leq \sqrt{1 - e^{-D_{KL}(\mathcal{G}_1(\mathcal{D})^n \; \| \; \mathcal{G}_0(\mathcal{D})^n)}} \tag{27}$$

$$= \sqrt{1 - e^{-nlog\left(\frac{1-\min(\alpha_0,\alpha_1)}{1-\max(\alpha_0,\alpha_1)}\right)}} \tag{28}$$

$$= \sqrt{1 - \left(\frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right)^n} \tag{29}$$

Since the function $f()$ is boolean, an adversary can choose to focus on a property value of 0 or 1, and infer the ratio of one using the other. Thus, any two ratios (α_0, α_1) can be alternatively seen as $(1-\alpha_0, 1-\alpha_1)$. Combining the two inequalities above and plugging them back in (11), we get an upper bound on accuracy:

$$\frac{1}{2} + \frac{\min\left\{\sqrt{1 - \left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)^n}, \sqrt{1 - \left(\frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right)^n}\right\}}{2} \tag{30}$$

Note that the proof of this bound hinges on both the distributions originating from the same underlying distribution $\mathcal{D}$, which is why we use $\mathcal{G}_0(\mathcal{D})$, $\mathcal{G}_1(\mathcal{D})$ instead of some arbitrarily defined distributions $\mathcal{D}_0$, $\mathcal{D}_1$.

A.2 Proof of Theorem 4.2

Consider Lemma 4.1: let ω be the observed distinguishing accuracy for some attack. Let n_{leaked} be the effective value of n corresponding to the given attack, *i.e.* Equating it with the best distinguishing accuracy for this value of n, we can compute n_{leaked}. If $\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)} \geq \frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}$:

$$2\omega - 1 = \sqrt{1 - \left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)^{n_{leaked}}} \tag{31}$$

$$\log(1 - (2\omega - 1)^2) = n_{leaked}\left(\log\left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)\right) \tag{32}$$

$$n_{leaked} = \frac{\log(4\omega(1-\omega))}{\log\left(\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}\right)} \tag{33}$$

Similarly, for the case of $\frac{\min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)} < \frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}$, we get:

$$n_{leaked} = \frac{\log(4\omega(1-\omega))}{\log\left(\frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right)} \tag{34}$$

Combining these two cases, we get:

$$n_{leaked} = \frac{\log(4\omega(1-\omega))}{\log(\max\left(\frac{min(\alpha_0,\alpha_1)}{\max(\alpha_0,\alpha_1)}, \frac{1-\max(\alpha_0,\alpha_1)}{1-\min(\alpha_0,\alpha_1)}\right))} \tag{35}$$

A.3 Proof of Theorem 4.3

Assume the adversary can fully recover a dataset S (of size N) from some model M trained on it. Let n_1 be the number of entries in S that are 1, and n_0 0 such that $n_0 + n_1 = N$. Then, the conditional probability density function of the underlying distribution $\mathcal{D}$ having $\Pr[1] = z$ (assume all z are equally likely), given the observed dataset S, can be written using the continuous Bayes' rule as:

$$\Pr[z \mid S] = \frac{\Pr[S \mid z]\Pr[z]}{\int_0^1 \Pr[S \mid x]\Pr[x]\,dx} \tag{36}$$

$$= \frac{\binom{N}{n_1} z^{n_1}(1-z)^{n_0}}{\int_0^1 \binom{N}{n_1} x^{n_1}(1-x)^{n_0}dx} \tag{37}$$

$$= z^{n_1}(1-z)^{n_0}\frac{\Gamma(n_0+n_1+2)}{\Gamma(n_0+1)\Gamma(n_1+1)} \tag{38}$$

The above conditional probability is maximized when $z = \frac{n_1}{N}$, i.e. the guessed ratio is the ratio observed in the given sample S. Then, we can compute the expected square error over all possible datasets of size N, given that the distribution they were sampled from has a proportion value α:

$$\mathbb{E}\left[(z-\alpha)^2\right] = \mathbb{E}\left[z^2\right] + \alpha^2 - 2\alpha\,\mathbb{E}[z] \tag{39}$$

We can then compute $\mathbb{E}[z]$ as:

$$\sum_{n_1=0}^{N} \frac{n_1}{N}\binom{N}{n_1}(\alpha)^{n_1}(1-\alpha)^{N-n_1} \tag{40}$$

$$= \frac{1}{N}\sum_{n_1=0}^{N} n_1\binom{N}{n_1}(\alpha)^{n_1}(1-\alpha)^{N-n_1} = \alpha \tag{41}$$

Similarly, $\mathbb{E}\left[z^2\right]$ can be computed as:

$$\sum_{n_1=0}^{N} (\frac{n_1}{N})^2\binom{N}{n_1}(\alpha)^{n_1}(1-\alpha)^{N-n_1} \tag{42}$$

$$= \frac{1}{N^2}\sum_{n_1=0}^{N} n_1^2\binom{N}{n_1}(\alpha)^{n_1}(1-\alpha)^{N-n_1} \tag{43}$$

$$= \alpha^2 + \frac{\alpha(1-\alpha)}{N} \tag{44}$$

Plugging (41) and (44) in (39), we get:

$$\mathbb{E}\left[(z-\alpha)^2\right] = \frac{\alpha(1-\alpha)}{N} \tag{45}$$

Thus, for an observed square error ω for some attack, n_{leaked} can be computed as:

$$n_{leaked} = \frac{\alpha(1-\alpha)}{\omega} \tag{46}$$

A.4 Proof of Lemma 4.4

We assume that both degree distributions follow Zipf's law, such that the PDF for either of $\mathcal{G}_0$ or $\mathcal{G}_1$can be written as

$$\rho_b(x) = \frac{x^{-s_b}}{H_{N_b,s_b}} \tag{47}$$

where $H_{N,s}$ is the N^{th} generalized harmonic number of order s, N_b corresponds to the maximum degree (with nonzero probability) $\mathcal{G}_B(\mathcal{D})$, and s_b determines the spread of the distribution. Since the inequality between the total variation distance and accuracy is independent of the underlying distributions, (11) applies in this case too.

Without loss of generality, let $N_1 \geq N_0$. In that case, the relative entropy from $\mathcal{G}_0$ to $\mathcal{G}_1$ would be undefined, since support($\mathcal{G}_1$) $\subseteq$ support($\mathcal{G}_0$) would be false. Computing the relative entropy from $\mathcal{G}_1$ to $\mathcal{G}_0$, we get:

$$D_{KL}(\mathcal{G}_0(\mathcal{D}) \parallel \mathcal{G}_1(\mathcal{D})) = \sum_{n=1}^{N_1} \rho_0(x)\log\left(\frac{\rho_0(x)}{\rho_1(x)}\right) \tag{48}$$

Since $\rho_0(x)$ only applies until N_0, it evaluates to 0 for $n > N_0$. Substituting:

$$\sum_{n=1}^{N_0} \rho_0(x)\log\left(\frac{\rho_0(x)}{\rho_1(x)}\right) + \sum_{n=N_0+1}^{N_1} 0\cdot\log\left(\frac{0}{\rho_1(x)}\right) \tag{49}$$

$$= \frac{1}{H_{N_0,s_0}}\left(\sum_{n=1}^{N_0} x^{-s_0}\left(\log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) + \right.\right. \tag{50}$$

$$\left.\left.(s_1-s_0)\log(x)\right)\right) \tag{51}$$

$$= \frac{1}{H_{N_0,s_0}}\left(H_{N_0,s_0}\log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) + \right. \tag{52}$$

$$\left.(s_1-s_0)\sum_{n=1}^{N_0} x^{-s_0}\log(x)\right) \tag{53}$$

$$= \log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) + \frac{s_1-s_0}{H_{N_0,s_0}}\sum_{n=1}^{N_0} x^{-s_0}\log(x) \tag{54}$$

If $s_1 > s_0$:

$$D_{KL}(\mathcal{G}_0(\mathcal{D}) \parallel \mathcal{G}_1(\mathcal{D}))$$

$$\leq \log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) + \frac{s_1-s_0}{H_{N_0,s_0}}\sum_{n=1}^{N_0} x^{-s_0}\log(N_0) \tag{55}$$

$$= \log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) + (s_1-s_0)\log(N_0) \tag{56}$$

If $s_1 \leq s_0$:

$$D_{KL}(\mathcal{G}_0(\mathcal{D}) \parallel \mathcal{G}_1(\mathcal{D})) \leq \log\left(\frac{H_{N_1,s_1}}{H_{N_0,s_0}}\right) \tag{57}$$

Computing the total variation distance for n samples according to (24) for both cases, we get an upper bound on $\delta(\mathcal{G}_0(\mathcal{D})^n, \mathcal{G}_1(\mathcal{D})^n)$ as:

$$\begin{cases} \sqrt{1-\left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}}N_0^{s_0-s_1}\right)^n}, & \text{if } s_1 > s_0 \\ \sqrt{1-\left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}}\right)^n} & \text{otherwise} \end{cases}$$

Plugging this in (11) to get an upper bound on the distinguishing accuracy.

$$\text{Accuracy} \leq \frac{1}{2} + \frac{\sqrt{1-\left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}}N_0^{(s_0-s_1)\mathbb{I}[s_1>s_0]}\right)^n}}{2} \tag{58}$$

A.5 Proof of Theorem 4.5

Consider Lemma 4.4: let ω be the observed distinguishing accuracy for some attack. Let n_{leaked} be the effective value of n corresponding to the given attack, *i.e.* Equating it with the best distinguishing accuracy for this value of n, we get:

$$\omega = \frac{1}{2} + \frac{\sqrt{1 - \left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}} N_0^{(s_0-s_1)\mathbb{I}[s_1>s_0]}\right)^{n_{leaked}}}}{2} \tag{59}$$

$$n_{leaked} = \frac{\log(4\omega(1-\omega))}{\log\left(\frac{H_{N_0,s_0}}{H_{N_1,s_1}}\right) + (s_0 - s_1)\mathbb{I}[s_1 > s_0]\log(N_0)} \tag{60}$$

B Leakage by Layers

Observing differences in performance when focusing on different network layers of the same model for CelebA (Section 6.3.1, Figure 11) raises an interesting question: *how does information leaked vary across model layers?* Understanding and identifying which layers leak the most information can help better understand the distribution inference risks and how to mitigate them, as well as how to make attacks more efficient. Here, we propose a simple test to help the adversary rank layers for value in distinguishing between the given distributions, and show how some layers (the first layer, in most cases) seem to capture properties of the training distribution better than others in a given model. Meta-classifier attacks are expensive and deciphering what they learn is challenging—identifying critical parameters can both improve understanding and lower resource requirements. If we can use just a fraction of the model's parameters, we may be able to achieve comparable inference performance with fewer shadow models and much lower meta-classifier training costs.

Identifying Useful Layers. Let j be some layer of the model for which the adversary wishes to gauge inference potential. We optimize query point $\hat{x}$ to maximize the difference in the total number of activations for layer j between models trained on datasets from the two distributions:

$$M_j(x) = \sum_i \mathbb{I}[(M[:j](x))_i > 0]$$

$$\hat{x} = \arg\max_x \left| \sum_{i;y_i=0} M_j^i(x) - \sum_{i;y_i=1} M_j^i(x) \right|, \tag{61}$$

where $M[:\ j](x)$ refers to the activations after layer j of model M on input x. The adversary can use a set of test points to select one that maximizes the above constraint. Then, similar to the process for Threshold Test (Equation 8), the adversary finds a threshold on the number of activations to maximize distinguishing accuracy. By iterating through all layers and computing the corresponding accuracies, the adversary can create a ranking of layers to estimate how much information these layers can potentially leak. This process is computationally much cheaper than running a meta-classifier experiment for all layers, and can be done with as few as 20 models. Once it has ranked all the layers, the adversary can pick the most informative ones (even just a single layer suffices in some cases) to train the meta-classifier. Since the resulting meta-classifier has fewer parameters (as it computes over fewer model layers), it can be trained using far fewer shadow models than when all network parameters are used, without having a significant impact on distinguishing accuracy.

Results. To understand how well the layer-identification process correlates with meta-classifier performance, we also perform experiments where each layer's parameters are used one at a time to train the meta-classifier. We run the layer-identification process, as described in Equation 61, for all layers across datasets. For the numbers reported in Table 3, the adversary samples data from its local test set to maximize Equation 61. Distinguishing accuracies reported in this table are on the adversary's models since it uses this ranking of layers to train a meta-classifier for its attack on the targeted model. For most cases, the layers closest to the inputs are identified as most useful. These accuracies for CelebA align with observations from previous experiments (Section 6.3) as well—for distinguishing sex ratios, the convolutional layers (until layer 5) seem to be more useful; for age, the fully-connected layers appear to be most useful. Layers of machine-learning models closer to the input are commonly associated with learning generic patterns, and later layers more abstract ones along with invariance to the given task [36]. Thus, the position of layers identified to be most useful is telling of how close the target property is to the input space or task.

Excluding the last layer does not lead to a significant performance drop. Intuitively, layers closer to the output will capture invariance for the given task and are thus less likely to contain any helpful information that prior layers would not already capture. If the last layer reveals enough information for the attack to succeed, then a black-box attack should also be possible. Results from layer-wise meta-classifier experiments con-

Dataset	Layer 1	2	3	4	5	6	7
CelebA (female)	**89.2**	76.7	75.8	74.2	70.0	66.7	63.3
CelebA (old)	64.2	60	60	68.3	**73.3**	70	66.7
Census (female)	**62.0**	58.0	56.4	-	-	-	-
Census (white)	**81.7**	75.0	63.3	-	-	-	-
RSNA Bone Age	**65.0**	64.0	-	-	-	-	-
ogbn-arxiv	93.3	95.0	**98.3**	-	-	-	-
Chord	69.4	60.0	64.0	64.8	57.2	50.2	-

Table 3. Maximum accuracy using layer-identification method. Since the last layer in all of these models is used for classification with a Softmax/Sigmoid activation, the process in Equation 61 cannot be applied to the last layer.

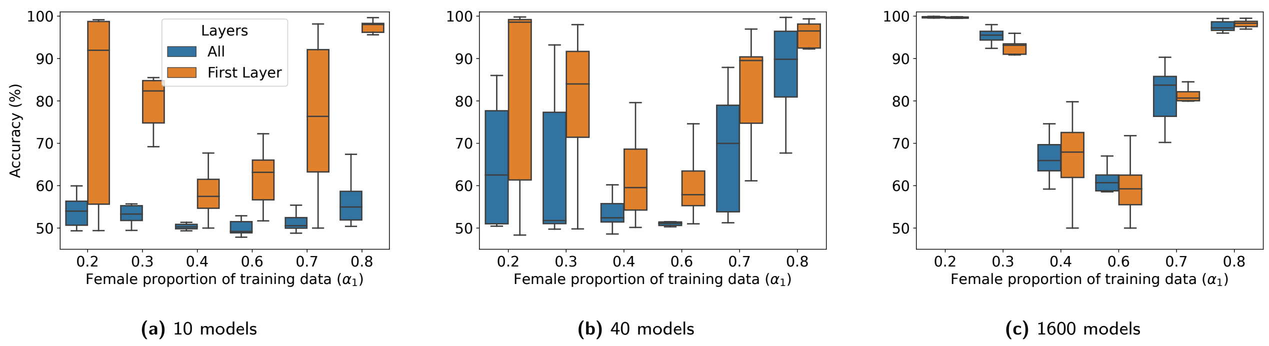


Fig. 7. Classification accuracy for distinguishing between models with different training distributions on the RSNA Bone Age dataset, for meta-classifiers trained with (a) 10, (b) 40, and (c) 1600 models. Orange box plots correspond to using parameters only from the first layer, while blue box plots correspond to using all (three) layers' parameters. Although the experiment that uses just the first-layer's parameters has much more variance, its average performance (and even the first quartile) is better than that of the version that uses all the layer's parameters.

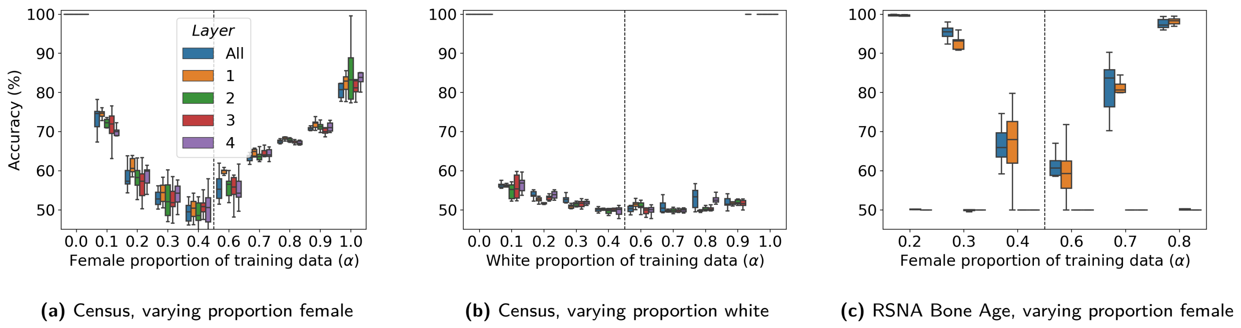
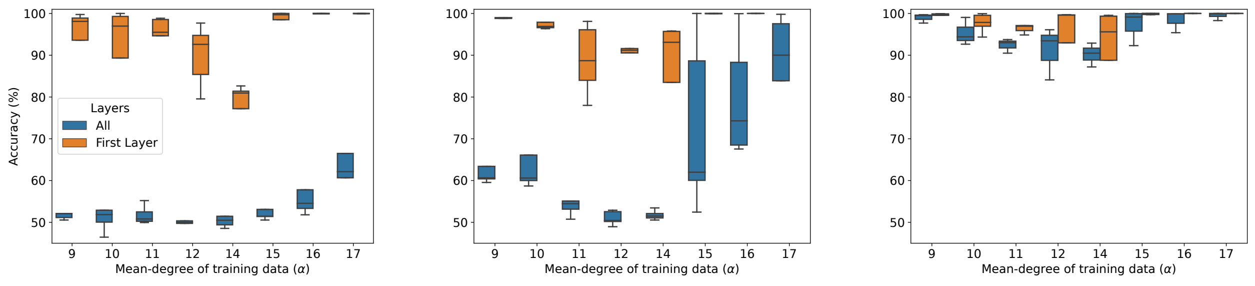


Fig. 8. Classification accuracy for distinguishing between training distributions for unseen models for on Census (sex: left, race: middle) and RSNA Bone Age, while varying the models' layers used while training meta-classifiers. There is no clear winner in the case of Census, while the first layer seems to the most useful for the case of RSNA Bone Age.

firm how the last layer's parameters rarely appear useful for distribution inference. Using these observations, we train meta-classifiers while using parameters only from some of the layers selected on the ranking we obtain via layer-identification experiments. We observe a clear advantage of doing so across all datasets, with minimal decreases in accuracy. For instance, using just the first layer produces a meta-classifier with only 20 training models on RSNA Bone Age (orange boxes in leftmost graph in Figure 7) that performs much better than using parameters from all of the layers. In order for the meta-classifier trained on all parameters to approach the accuracy of the one-layer meta-classifier, hundreds of shadow models are needed.



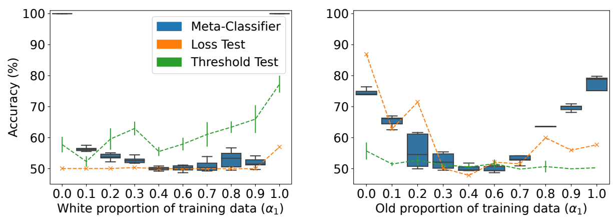
(a) 20 models **(b)** 100 models **(c)** 1600 models

Fig. 9. Classification accuracy for distinguishing between models with different training distributions for on the ogbn-arxiv dataset. Left to right: meta-classifiers trained using 20, 100, and 1600 (original experiment) models, respectively. Orange box plots correspond to using parameters only from the first layer, while blue box plots correspond to using all (three) layers' parameters. Using as few as 20 models is sufficient for satisfactory meta-classifier performance when the right layers are identified and used.

For datasets like Census, all the layers seem to equally useful while for RSNA Bone Age, only the first layers' parameters are useful (see Figure 8). When using the first layer's parameters, the adversary can achieve an average of 75% accuracy with as few as 20 models, compared to 54% when using the entire model. In fact, the first layer is identified as most useful and using any other layer leads to near-random performance. Additionally, for larger models like those for CelebA, the adversary can pick more than one layer—using as few as three layers of the model can help lower computational resources. As observed in ablation experiments with convolutional and linear layers for CelebA (old people), using just the last three layers (of which two the layer-identification process identifies), the adversary can train its meta-classifiers while using significantly fewer models. When using 100 models to train the meta-classifier, using just the fully-connected layers gives a 4% absolute improvement in accuracy, along with 0.5% reduction in standard deviation across experiments.

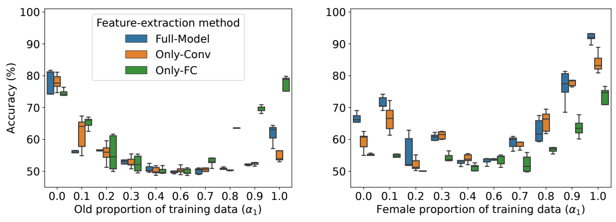
Graph Datasets. The layer-identification process does not work on the graph datasets. It incorrectly predicts the third layer as most useful for ogbn-arxiv, whereas actual performance with that layer's parameters leads to a significant performance drop. We suspect this behavior can be explained by the inherent properties of the graph data. Intermediate activations for nodes can have complicated interactions with neighboring nodes, leading to the detection method's instability when analyzing activation values. These challenges on graph datasets is something that we plan to investigate in future work. Nonetheless, the fact the first two layers are useful for both graph datasets suggests a useful direction for graph-based property inference attacks.

C Additional Figures



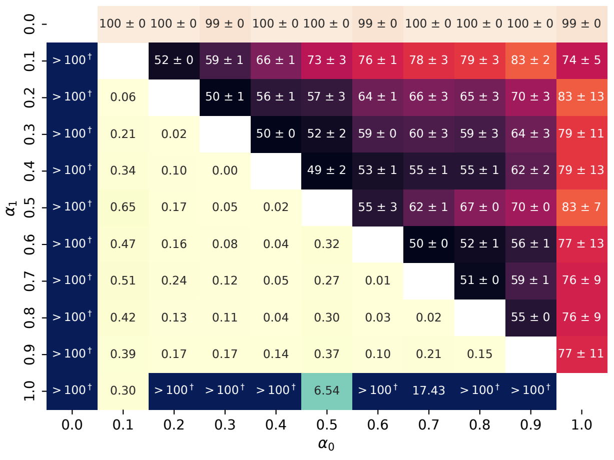
(a) Census, race **(b)** CelebA, age

Fig. 10. Distinguishing accuracy for proportion of (a) whites in training data for the Census and (b) old people in the CelebA. The black-box attacks approach the performance of white-box meta-classifier attacks for some distributions (especially the Threshold Test and the Census (race) task).

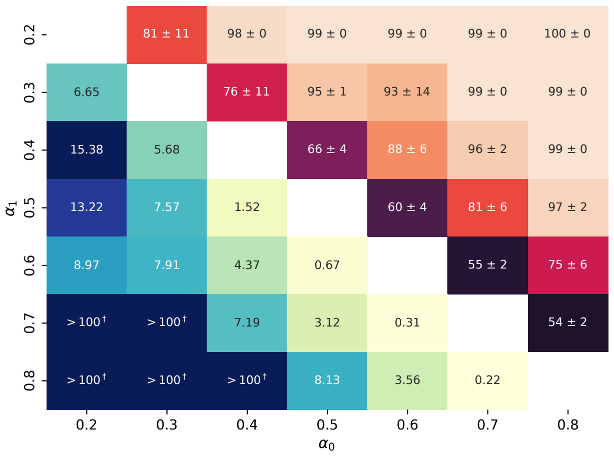


(a) CelebA, age **(b)** CelebA, females

Fig. 11. Distinguishing accuracy for meta-classifiers for proportion of (a) old people and (b) females in the training data for CelebA. Using all the layers' parameters is not necessarily helpful and can lead to lower performance (e.g., CelebA, females).

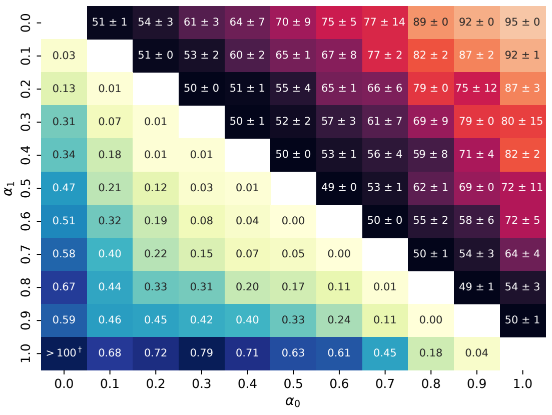


(a) Census

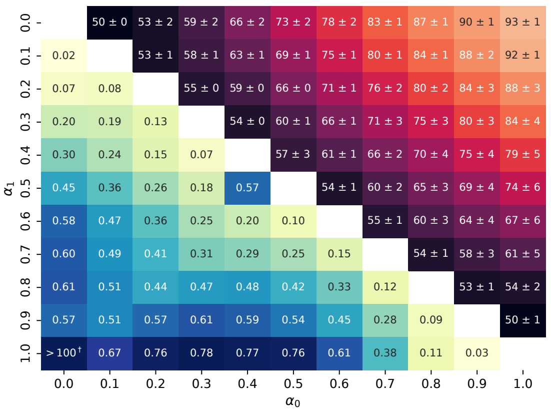


(b) RSNA Bone Age

Fig. 12. Effectiveness of meta-classifiers in distinguishing proportions of females on Census and RSNA Bone Age. The bottom-left triangles of the heatmaps show the n_{leaked} values, and the top-right triangles show the distinguishing accuracies between training distributions $\mathcal{G}_0(\mathcal{D})$ with ratio α_0 and $\mathcal{G}_1(\mathcal{D})$ with ratio α_1 for females. Distinguishing accuracies seem to follow intuitive patterns, with an increase as the distributions diverge (larger $|\alpha_0 - \alpha_1|$). The n_{leaked} values allow for comparisons of attack power between different pairs of distributions, but also show that very little leakage is observed for most settings, except for RSNA Bone Age.



(a) Binary Meta-Classifiers



(b) Regression Meta-Classifier

Fig. 13. Distinguishing binary ratio properties using (a) binary classifiers and (b) regression meta-classifiers, for CelebA (age). n_{leaked} values (lower triangle) and classification accuracies (upper triangle) for distinguishing proportion of old people (ratios α_0, α_1) in training data for the CelebA dataset. n_{leaked} with binary meta-classifiers lower, especially for cases where $|\alpha_0 - \alpha_1|$ is small.