Dissecting Distribution Inference

Anonymous Authors

***Abstract*—A distribution inference attack aims to infer statistical properties of data used to train machine learning models. These attacks are sometimes surprisingly potent, but the factors that impact distribution inference risk are not well understood and demonstrated attacks often rely on strong and unrealistic assumptions such as full knowledge of training environments even in supposedly black-box threat scenarios. To improve understanding of distribution inference risks, we develop new black-box and white-box attacks that consistently outperform the current state-of-the-art. Our new black-box attack even outperforms the best known white-box attacks in most settings. Using these new attacks, we evaluate distribution inference risk while relaxing a variety of assumptions about the adversary's knowledge under black-box access, like known model architectures and label-only access. Finally, we evaluate the effectiveness of previously proposed defenses and introduce new defenses. We find that although noise-based defenses appear to be ineffective, a simple re-sampling defense can be highly effective.**

I. Introduction

Machine learning models are susceptible to several disclosure risks, including leaking sensitive information related to training data. Distribution inference considers what a model reveals about its entire underlying training data, in contrast with inference attacks that focus on individual records like membership inference and memorization attacks. Initial work focuses on inferring ratios of data with a particular attribute, referring to the attacks as "property inference". Examples include inferring the accent of speakers in voice recognition models [1], targeting ratios of characteristics like gender labels [2], and estimating sentiment across email datasets [3]. More recently, there have been attempts to extend these attacks to properties beyond ratios, such as predicting graph density [4] and direct regression over graph mean-degree [5] with successful inference adversaries with just black-box access.

State-of-the-art distribution inference attacks achieve non-trivial distinguishing accuracies [2], [5], [6] and thus pose a privacy risk, but the actual amount of leakage achieved is often minimal. Leakage varies significantly across different datasets, but for most settings the best current attacks leak no more information than what one or two samples from the distribution would reveal [5]. This is enough to have a significant advantage in distinguishing highly dissimilar distributions, but seems unlikely to pose a serious privacy risk in most cases.

Contributions. We advance the understanding of distribution inference risk on several fronts including improved attacks, analysis of risk, and development and evaluation of defenses. To enable better estimation of this risk, we introduce new attacks that can significantly outperform state-of-the-art attacks. We introduce a new black-box attack, the KL Divergence Attack, that uses distributional similarity in predictions (Section II-B). We also propose a new white-box attack, the Affinity Graph Attack, that uses the similarity between intermediate model activations across data to predict training distributions (Section II-C). These attacks substantially outperform the current state-of-the-art (Section III), increasing previous estimates of inference leakage for various datasets in the literature. Surprisingly, we find that in most settings our black-box KL Divergence Attack is more effective than the best known white-box attacks (including the Affinity Graph Attack). We also evaluate the black-box attacks in more realistic settings where the adversary does not have as much information as is typically assumed in inference experiments (Section IV). We evaluate the impact of different model architectures (Section IV-A), the lack of common feature extractors (Section IV-B), and relaxing the assumption of prediction probabilities to label-only access settings (Section IV-C). Our experiments find large variances in inference risk due to relaxing assumptions about model architectures and feature extractors, but demonstrate that attacks can be effective in label-only settings.

Section V evaluates defenses, including both previously proposed ones and new ideas. Most privacy-related defenses for machine learning involve adding noise at some stage of the training; for instance, at the gradient-level in differentially private training [7], or at the data level with most implementations of adversarial training [8]. Our experiments find that these noise-based defenses provide little mitigation against distribution inference (Section V-A). At some level, this is unsurprising since these defenses are designed to protect individual training records, not distributional properties, and we show a connection between how well a model generalizes to the task distribution and its susceptibility for distribution inference leakage (Section V-B). We then develop a simple and inexpensive mitigation based on data re-sampling (Section V-C) which can protect against distribution inference in most cases where an adversary knows the statistical property to protect.

II. Attacks

Before introducing our attacks, we summarize the formal definition of distribution inference we use (Section II-A). Then, we describe previous and our new attacks in the black-box (Section II-B) and white-box (Section II-C) settings.

A. Defining Distribution Inference

The goal of distribution inference refers is to infer sensitive properties of the distribution used to train a machine learned model given some level of access to that model. Mahloujifar et al. [3] formalized a notion of distribution inference that can capture the proportion of a dataset satisfying some property. They distinguish two distributions, $\mathcal{D}_+$ and $\mathcal{D}_-$, corresponding

to data with some attribute (linked to the property, such as 'female') value, and define a mixture of these distributions for some α corresponding to the probability of sampling a record from $\mathcal{D}_+$. Suri et al. [5] generalize this notion by using distribution transformation functions $\mathcal{G}_0$ and $\mathcal{G}_1$ to transform an underlying distribution $\mathcal{D}$. When considering proportional properties, the transformers can correspond to sampling records with a given attribute value for any two valid probabilities, α_0 and α_1. We follow this formalization, since this is more generic and better captures assumptions like access to data from some underlying distribution $\mathcal{D}$. The adversary's task in this setting thus corresponds to distinguishing models trained on data from $\mathcal{G}_0(\mathcal{D})$ and $\mathcal{G}_1(\mathcal{D})$, given access to a dataset sampled from $\mathcal{D}$ and some level of access to the trained model.

B. Black-Box Attacks

Black-box attacks assume the adversary has the ability to submit inputs to the trained model and observe the response but does not have direct access to the model. In addition, the adversary has access to some representative data from some distribution $\mathcal{D}$, and seeks to infer which of the transformations, $\mathcal{G}_0$ or $\mathcal{G}_1$, corresponds to the victim's training distribution. Using knowledge of $\mathcal{D}$ and the transformation functions $\mathcal{G}_0$ and $\mathcal{G}_1$, the adversary is able to train shadow models locally. Knowledge of the candidate distributions is necessary to be able to distinguish between them, and it is reasonable to assume an adversary with enough computational resources to train models locally. Most research assumes that the victim and adversary use the same model architecture (e.g., [6], [9]), and that the adversary has access to model prediction confidence vectors (e.g., [6], [10], [11]). In Section IV, we consider settings where the adversary has less knowledge of the victim model and the model API only outputs labels. Next, we review previous black-box distribution inference attacks, and then introduce our new KL Divergence Attack.

Prior Work. Zhang et al. [6] propose meta-classifier attacks that use probability vectors from models for a specific set of query points. Similar ideas are explored in related tasks [10]. Suri et al. [5] propose the Loss Test and Threshold Test attacks that compare model accuracies on candidate distributions to predict training distributions. The Threshold Test performs best of these, and we use it in our experiments. The Threshold Test uses locally trained models to derive a threshold on observed accuracy on a given data sample, which is then used to predict the training distribution of a model. This attack yields non-trivial inference accuracies in many cases but falls short of the white-box attacks by huge margins for most settings. The only previous attack designed to work with label-only predictions is Mahloujifar et al.'s attack in a setting where active adversaries that can poison the victim's training data [3].

KL Divergence Attack (KL). Recent work by Hartley et al. [12] demonstrates how the presence of unique features, even if present in one record of training data, can impact output probability distributions. Motivated by their use of KL divergence to differentiate between the two scenarios (instance present or not), we propose an attack that compares the KL divergence in output probabilities of the victim model, using local models M_0, M_1 trained from training distributions $\mathcal{G}_0(\mathcal{D})$, $\mathcal{G}_1(\mathcal{D})$ respectively. To estimate these KL divergence values, the adversary uses a local sample of models:

$$\mathbb{E}[D_{KL}(N \parallel M)] = \mathbb{E}\left[\sum_{c \in \mathcal{C}} N(x)_c \log\left(\frac{N(x)_c}{M(x)_c}\right)\right] \quad (1)$$

where $M(x)_c$ corresponds to the prediction probability corresponding to class c (out of all classes $\mathcal{C}$) for some point x for model M, and the expectation $\mathbb{E}[]$ is taken over the adversary's data. The adversary then defines a "weighted vote" for a pair of models (N, P) with respect to M:

$$\lambda(M, N, P) = \mathbb{E}[D_{KL}(N \parallel M)] - \mathbb{E}[D_{KL}(P \parallel M)]. \quad (2)$$

A positive quantity $\lambda(M, N, P)$ thus indicates that the model M has its predictions distributed closer to P than N. Using its collection of local models trained on the two candidate distributions, the adversary then computes and aggregates this "weighted vote" across all pairs of its local models (M_0^i, M_1^j):

$$\hat{b} = \mathbb{I}\left[\sum_i \sum_j \lambda(M, M_0^i, M_1^j) > 0\right] \quad (3)$$

The rule above thus effectively checks all its pairs of local models and compares similarities in prediction distributions with a given victim model.

C. White-Box Attacks

In the white-box setting, the adversary additionally has direct access to the victim's model including its trained parameters. Although this access model assumes a stronger adversary, it is a realistic adversary for many scenarios, like when models are deployed on client devices. It is also useful in two ways: 1) gauging the extent of inference leakage, helping bound risk and understand it better, and 2) studying patterns and trends across properties and models to help better understand distribution risk and come closer to inventing effective defenses.

Prior Work. The main previous white-box distribution inference attacks are based on permutation-invariant networks [2]. They look at model parameters across all layers of a multi-layer perceptron to generate a feature representation for the entire model that is insensitive to arbitrary reorderings of neurons. These attacks, originally designed for networks with linear layers, have been extended to support convolutional layers [5]. Other distribution inference attacks in the literature follow a similar meta-classifier approach: using parameter extraction for support vector machines [1] or using model gradients [9].

Affinity Graph Attack. Instead of looking at model activations directly, we propose a technique that focuses on the relationship between data across models. The attack starts with p data-points π, $|\pi| = p$. For a given model m, we pass each data point π_i through the model, collecting intermediate feature activations l_i^j after each layer j. Then, for each layer j we compute the cosine similarity between every point π_i and π_k: $cos(l_i^j, l_k^j)$.

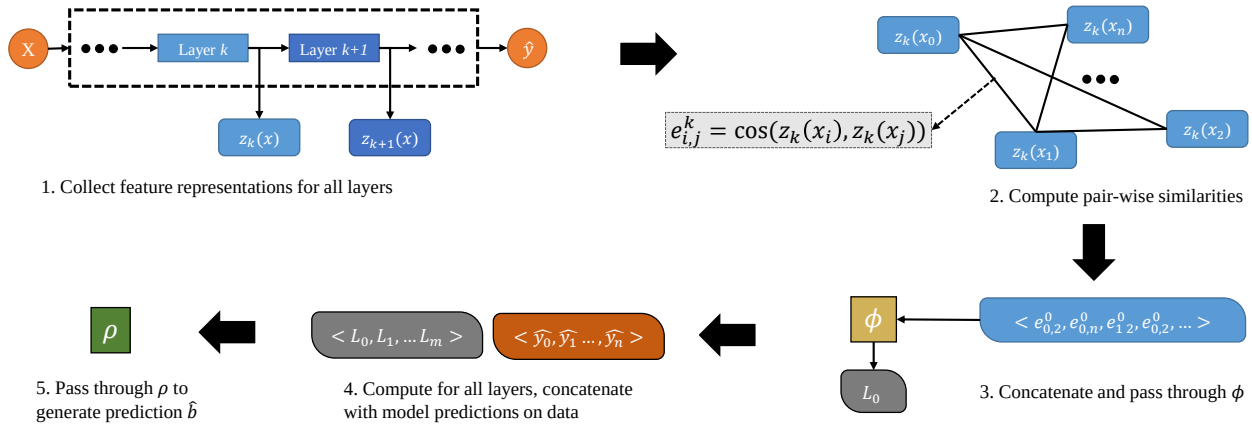

Fig. 1: Flow diagram for the Affinity Graph Attack classifier. For each model, latent features are collected for all datapoints (1). Then, pair-wise cosine similarities of the features are computed (2) and processed layer-wise (3), leading to a single representation for the given model (4). A linear classifier can then be used to generate predictions (5).

This is repeated for every $\binom{p}{2}$ pair of data-points, yielding a list of cosine similarity scores per layer. Since $\binom{p}{2}$ gets large very quickly, we train a small neural network model ϕ to learn a smaller representation of these lists of similarity scores. These scores are concatenated and passed through the model ϕ, for layer j:

$$L_j = \phi\big(\cos(l_0^j, l_1^j),\ \ldots,\ \cos(l_0^j, l_p^j),\ \ldots,\ \cos(l_{p-1}^j, l_p^j)\big).$$

This process is repeated for all layers in the model and these layer-wise features are then concatenated and passed through a linear layer ρ, along with model predictions, to obtain a final prediction via the meta-classifier:

$$\hat{b} = \rho(L_0,\ L_1,\ \ldots,\ L_m,\ \hat{y}_0,\ \hat{y}_1,\ \ldots,\ \hat{y}_n).$$

Figure 1 illustrates our method. Note that the network ϕ is shared across all layers. Since this meta-classifier is designed to learn how features for various data changes across layers, having a shared ϕ model helps the meta-classifier learn the desired inference task easier, in addition to lowering the number of trainable parameters. This attack is both permutation- and scale-invariant by design, since the cosine similarity function is also scale and permutation invariant. Additionally, while PIN is applicable only to feed-forward neural networks and CNNs, AGA works with any model architecture from which intermediate features can be extracted, including models with skip-connections like ResNets [13] and RNNs [14]. Instead of trying to model inter-layer relationships by explicit concatenation of prior layers (like permutation-invariant networks), layer-wise representations are simply concatenated, and the final model ρ can learn relationships between layers.

Combined Attack. While white-box and black-box attacks may perform well individually, an adversary with white-box access may be able to obtain the best performance by combining both attacks. We use a decision tree to combine the KL Divergence Attack and Affinity Graph Attack attacks. To train the decision tree, the adversary takes a small sample of local models (that it trained for the meta-classifier) and generates prediction scores using the Affinity Graph Attack and KL Divergence Attack. Then, it concatenates scores from both attacks per model to obtain a tuple of scores, which it then uses as features to train a Decision Tree classifier. We set the maximum depth of this decision-tree to be 2 to avoid overfitting. This combined attack is abbreviated as AGA+KL.

III. Results

We evaluate our proposed attacks on several datasets, including both established benchmarks used in prior work and new configurations and property-task combinations previously unexplored in the literature. Code for reproducing our experiments is available at https://anonymous.4open.science/r/dissecting_distribution_inference-0B2F/ (anonymized link for reviewing, will be in public github repository).

Our new attacks are significantly more potent than the previous state-of-the-art. For the white-box setting, Affinity Graph Attack outperforms Permutation Invariant Networks across nearly all experimental setups, and our KL Divergence Attack outperforms the Threshold Test by huge margins (Section III-C). Even more interestingly, the KL Divergence Attack, with only black-box access, outperforms Affinity Graph Attack by a large margin in nearly all settings. We study trends between the correlation of the task and property, and its impact on inference risk (Section III-D).

A. Datasets and Models

We evaluate our attacks on twelve task-property pairs across five datasets, summarized in Table I. These were selected to directly compare results with previous works (RSNA Bone Age, ogbn-arxiv, CelebA), to study the impact of task-property correlation on inference risk (various property-task pairs for CelebA), and to include datasets representing real-world use-cases, like Census19 and Texas-100X.

Census19 [15] is an updated and expanded version of the Adult Census dataset [16] based on data from the US Census

Dataset	Task/Property	Black-Box Attacks		White-Box Attacks		
		TT [5]	KL	PIN [2]	AGA	AGA+KL
Census19	Income/Females	1.0	**7.6**	1.8	2.0	6.3
	Income/Whites	0.1	**5.0**	1.6	0.8	2.9
Texas-100X	Income/Females	<0.1	<0.1	<0.1	0.5	<0.1
	Income/Whites	<0.1	0.1	0.1	0.1	0.1
	Income/Hispanic	<0.1	0.1	<0.1	0.2	0.2
CelebA	Mouth Open/Wavy Hair	<0.1	0.2	0.1	<0.1	<0.1
	Smile/Females	0.3	**2.8**	0.3	0.7	0.9
	Gender/Young	<0.1	<0.1	0.2	0.4	0.4
	Mouth Open/High Cheekbones	<0.1	**1.9**	0.8	0.6	1.0
RSNA Bone Age	Age/Females	<0.1	**3.0**	3.7	**7.8**	5.9
	Females/Age	0.3	**22.8**	5.0	6.3	3.7
ogbn-arxiv	Node classification/Mean Degree	<0.1	**220.4**	158.3	–	–

TABLE I: Effectiveness of inference attacks. Mean n_{leaked} values across varying α_1 values are reported, with fixed $\alpha_0 = 0.5$, using median performance of 5 trials over each attack, for KL Divergence Attack (KL), Affinity Graph Attack (AGA), and the combined AGA+KL attack. We also compare our results with the best prior attacks: Threshold Test (TT) [5] and Permutation Invariant Networks (PIN) [2]. For each setting where non-trivial (> 1.0) leakage is observed, the results for the most effective black-box and unrestricted attack is bolded. Due to computational constraints, and the significantly superior performance of KL Divergence Attack compared to Permutation Invariant Networks, we do not implement the Affinity Graph Attack for ogbn-arxiv.

Bureau. It contains a mixture of numerical and categorical features, and the same prediction task. We focus on the ratio of whites (race) and females (sex) as properties, and use a two-layer feed-forward neural-network as the architecture.

Texas-100X [17] contains demographic and medical information for patients across hospitals. The original dataset uses 100 possible classes for surgical procedure prediction. We slightly modify the task and focus only on data from the top 20 classes, reducing it to a 20-class classification task. We focus on the ratio of whites (race), females (sex), and Hispanics (ethnicity) as properties, and use a two-layer feed-forward neural-network.

CelebA [18] contains collections of face images of celebrities. Each image is annotated with attributes. We use three different tasks: smile detection, gender prediction, and mouth-open prediction. We conduct experiments with a convolutional neural network trained from scratch for this dataset, with five convolutional layers and pooling layers followed by three linear layers, which is the smallest network we could find with reasonable task accuracy. For our experiments with feature extractors, we also conduct experiments where the adversary uses a pre-trained FaceNet [19] model trained on the CASIA-WebFace [20] dataset, with a two-layer network. It leads to a drop in performance (from $\sim 92\%$ to $\sim 82\%$), but the point of such an experiment is indeed to assess inference risk in more practical settings. For the attack inference properties, we use the proportion of females (smile-detection task), old people (gender-prediction task), people with wavy hair (mouth-open-prediction task), and people with high cheekbones (mouth-open prediction task). These pairs are useful in comparing results with previous works, and also help cover a spectrum of different correlations between the task and property attributes.

RSNA Bone Age [21] contains x-ray images of hands, and the standard task is to predict the patient's age in months. We convert the task to binary classification based on an age threshold (> 132 months), and focus on the ratios of the females (available as metadata) as properties. We also consider a flipped scenario, where the task is to predict females, with the ratios of people below the age threshold as properties. We use a pre-trained DenseNet [22] model for feature extraction, followed by a two-layer network for classification. Similar to CelebA, we consider a setting where the adversary uses pre-trained feature extractor, while the victim trains models from scratch. Additionally, we also consider a setting where both the victim and adversary use the same feature extractor, but use different model architectures on top of the feature extractor

ogbn-arxiv [23] is a directed graph of citations between computer science arXiv papers, with the task as predicted subject area categories (out of 40) using features extracted from paper documents. We infer the mean node-degree property of the graph, and use Graph Convolutional Networks [24].

B. Experimental Setup

We build upon the experimental setup described in earlier works on distribution inference, using implementations and trained models provided by previous works [2], [5]. We include the datasets and victim/adversary splits used in previous experiments, and include results on two new datasets. For all of the experiments, we follow the processing pipeline described in Suri et al. [5] to obtain non-overlapping splits.

We perform each experiment five times and report mean values with standard deviation in all of our experiments. For each dataset, we train 250 victim models per distribution. For all black-box attacks, the adversary trains and uses 50 models per distribution for each trial. For white-box attacks, the adversary trains 800 models per distribution, of which 750 are used for training and 50 as the validation set. For cases with very large models (like DenseNet trained from scratch for RSNA Bone Age), we use 100 victim models per distribution.

KL Divergence Attack. Since using all pairs of adversary's models can be expensive, the attack uses a set fraction (0.8) of randomly chosen pairs to compute the expectation in Equation 3. We experiment with multiple values of this fraction, and observe comparable performance. For each pair of local models, the attack collects the difference in KL values. These differences are then normalized across all differences observed for local models, after which the adversary uses voting-based aggregation to generate the final prediction.

Affinity Graph Attack. Similar to the KL Divergence Attack, we use a fraction (0.2) of randomly chosen pairs of points. Since a given model can have multiple layers, we use specific layers for each model while training the meta-classifiers, which are fixed across all model architectures and were selected before the experiments were evaluated (based on computational constraints), making them free of selection bias.

AGA+KL. The adversary uses 100 models out of all its models for generating predictions with the AGA and KL, which it then uses to train a Decision Tree classifier on prediction probability scores from both attacks. Then, for any given victim model, the adversary runs AGA and KL and then feeds concatenate probability predictions from these two attacks to the Decision Tree to generate a final prediction.

C. Results

Table I summarizes the results of our distribution inference experiments. Directly comparing distinguishing accuracies across different distributions can be misleading, since certain pairs of distributions might be easier to distinguish than others. To standardize comparisons, we use the n_{leaked} [5] metric in our experiments. n_{leaked} values measure how much the inference attack leaks about the distributional property in a way that allows comparison across experiments. It can be interpreted as the number of records an adversary would need to sample directly from the training distribution to expect to make equally accurate predictions.

We vary α_1 and set $\alpha_0 = 0.5$ for the case of ratio-based properties, where a certain α value for a distribution means datasets sampled uniformly at random would have α fraction of the data with the property attribute 1, for e.g. ratio of females. For ogbn-arxiv, we use $\alpha_0 = 13$ and vary α_1 in the range of integers $[9, 17]$.

The majority of experimental evaluations in the literature follow a binary classification scenario, where the adversary is supposed to distinguish between two potential training distributions with property values α_0 and α_1. Although regression-based adversaries have been demonstrated [5] as being strictly more powerful, they are much more computationally expensive and we leave evaluations in that scenario to future work.

Trends across datasets. Inference leakage varies significantly across different datasets, with very little leakage for most cases in Texas-100X, substantial leakage for Census19, and exceptionally high leakage for the graph-based ogbn-arxiv dataset. The lack of virtually any inference risk in Texas-100X is surprising, as the features contain the property label, and data splits are processed per hospital during generation, making the victim and adversary distributions highly similar. This difference in inference risk between Census19 and Texas-100X, despite both being tabular datasets, reveals how just the nature of data (tabular, images) does not by itself determine inference risk and risk can vary unpredictably (at least based on current understanding) with aspects of the data. As previously observed by Suri et al. [11], leakage is quite high for RSNA Bone Age. Our new improved attacks identify vulnerable datasets, such as Census19, that would have been considered low leakage risks using previous state-of-the-art attacks.

	Number of Shadow Models			
Dataset/Task	5	10	25	50
Census19 (Sex)	7.1	7.4	7.6	7.6
Census19 (Race)	3.2	3.7	4.1	5.0
RSNA Bone Age (Age)	5.8	6.9	10.2	22.8
CelebA (Sex)	1.0	1.1	2.5	2.8

TABLE II: Impact of number of shadow models. Values are n_{leaked} for KL Divergence Attack. Even with only 5 models, the adversary is able to achieve non-trivial inference.

Comparing black-box attacks. The KL Divergence Attack outperforms Threshold Test in all cases with large margins. Across all of the settings, Threshold Test never achieves n_{leaked} values about 1.0 (indicating that the observed leakage is less than what an adversary would learn by sampling a single record from the training distribution), whereas the KL Divergence Attack produces meaningful leakage for all of the datasets besides Texas-100X. The superiority of the KL Divergence Attack can be attributed to the use of pairs of local models and their trends (which grow in the order $\binom{n}{2}$ for n models), as opposed to using information from models in isolation in the Threshold Test.

Number of shadow models. The black-box attacks use 50 shadow models per training distribution. We vary this number to 1) get an empirical lower bound on the number of shadow models required to achieve non-trivial leakage, and 2) study increase in information leakage with an increase in shadow models. Leakage is significant with only five shadow models per distribution and in most cases, and improves with more local shadow models (Table II).

White-box attacks. Our proposed white-box Affinity Graph Attack (AGA) outperforms the current state-of-the-art Permutation Invariant Networks (PIN) in all experimental settings that exhibit non-trivial leakage except for Census19 (Income/Whites) (but both attacks do much worse than KL). While PIN extracts signals from model parameters, AGA utilizes trends in model activations, and is thus able to better capture useful trends between data and intermediate model activations. The input feature space for AGA is smaller too, since it uses cosine similartity values over small samples of data, as opposed to full model parameters in PIN.

The black-box KL Divergence Attack performs surprisingly

well despite the weaker threat model, out-performing the best white-box attack in nearly all experimental settings. The one exception is RSNA Bone Age (Age/Females), where both white-box attacks outperform KL, and AGA leaks more than twice as much as PIN. Since an adversary in the white-box setting has access to more information than just the data and model predictions, it should be at least as powerful as a black-box adversary. We attribute the relative inffectiveness of the white-box attacks to two main reasons. First, although AGA utilizes model predictions, it does so differently from KL, as they are directly used as features for the meta-classifier, unlike comparisons in model prediction distributions in the latter. Secondly, AGA and other white-box attacks have a larger feature space and learning meta-classifiers additionally requires learning to recognize relevant patterns in model parameters, a huge and complex data distribution. The black-box attacks, on the other hand, are agnostic to parameters in the victim model and thus much easier to scale, resulting in better performance.

Our attempts at combining the white-box and black-box attacks do not work well. Although incorporating information from KL boosts performance of AGA, AGA+KL performs worse than KL in nearly all cases. An adversary is thus better off just using the KL Divergence Attack.

D. Correlation

The impact of correlation between the underlying task of a model and the property of its training distribution being inferred has been touched upon briefly in the literature [6], but not studied extensively. Intuition suggests there should be some linear relationship between inference risk with task-property correlation, but prior studies do not evaluate inference risk across a range of property-task correlations. We carefully pick pairs of properties and tasks for the CelebA dataset, such that there is a good range of correlations.

We conduct experiments with property correlations of ≈ 0 (Mouth Slightly Open–Wavy Hair), ≈ 0.14 (Smiling–Female), ≈ 0.28 (Female–Young), and ≈ 0.42 (Mouth Slightly Open–High Cheekbones). Across this range of correlations, average n_{leaked} values are 0.2, 2.8, <0.1, 1.9 as correlation values increase. The lack of any clear trend between correlation and inference risk is consistent with observations in the literature around task-property correlation and inference risk [6].

We perform similar analyses for the RSNA Bone Age dataset, where we flip the property and task. In this case, the correlation between the task and property remains the same, thus helping identify potential changes in inference leakage arising purely from the choice of the property itself. While switching from Age–Females to Females–Age, we observe a huge bump in n_{leaked} values: from 3.0 to 22.8. Although the choice of property and task are expected to impact inference risk, our results suggest that this choice may be more relevant to evaluating inference risk than property-task correlation itself.

IV. Impact of Adversary's Knowledge

Research on inference privacy typically considers threat models with one of two simplistic adversarial assumptions:

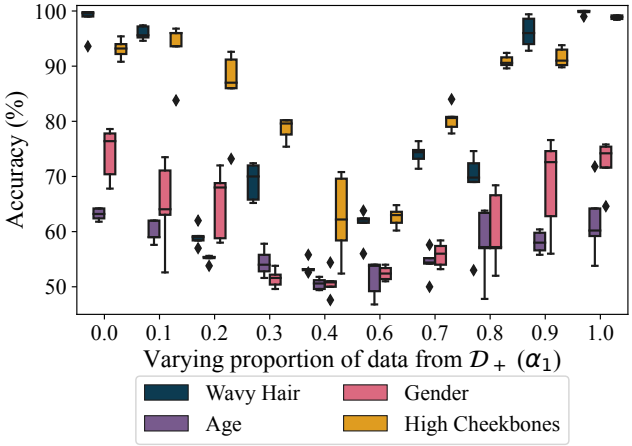

Fig. 2: Distinguishing accuracy for different task-property pairs for CelebA with varying correlation, for KL Divergence Attack.

white-box settings, where the adversary has full access to the model; and black-box settings, where the adversary has only API access to the model but receives full confidence vectors for each prediction and has complete knowledge of aspects of the training process and model architecture. The specific information available to an adversary in the black-box setting, however, may have a significant impact on inference risk. For instance, access to labeled data (for attacks) with prediction probabilities is often implicit, as is the use of the same model architectures and feature extractors between the victim and adversary. We study the impact of these common assumptions, and how relaxing them impacts inference risk. We measure impact on risk when the victim and adversary use different model architectures (Section IV-A), do not share feature extractors (Section IV-B), and when the available model API only provides label predictions (Section IV-C). Inference risk is somewhat robust to differences in model architectures, as long as the victim and adversary's models have similar capacity. The absence of shared feature extractors reduces inference risk significantly, but we find attacks can still succeed when only label predictions are available.

A. Model Architecture

In the white-box setting an adversary can directly observe the target model's architecture, but in black-box settings it is unrealistic to assume the adversary knows the target model architecture. Likely model architectures may be limited in certain domains like image data, where the victim is likely to use a popular model architecture such as DenseNet [22]. But a variety of models like random forests, support vector machines, and clustering-based classifiers can be used for tabular data and may even be picked by model trainers via automated tools [25].

Differences in victim and adversary architectures have not been previously explored, except by Mahloujifar et al. [3] for poisoning-based adversaries. In their setting, the victim and adversary can have different model architectures—the adversary uses logistic regression while the victim can use a variety of

Victim Model	Adversary Model RF	LR	MLP_2	MLP_3
Random forest (RF)	7.9	5.1	10.6	13.3
Linear regression (LR)	16.6	25.9	25.9	25.9
Two-layer perceptron (MLP_2)	5.4	7.2	7.6	6.3
Three-layer perceptron (MLP_3)	7.2	7.5	2.5	3.2

TABLE III: Variation by model type. Each values is observed n_{leaked} from KL attack for Census19 (Sex), for different combinations of model types for victim and adversary.

different feed-forward neural networks. They note a drop in inference risk with an increase in victim model complexity. It is thus unclear if these trends are specific to the model architecture themselves. Additionally, the adversary's model architecture is kept the same, so they did not explore the potential for higher inference risk with better local models.

To identify trends in inference risk with differences between architectures, we train multiple models with different architectures for both the victim and adversary. For Census (Gender), we try all possible combinations out of linear regression (LR), multi-layer perceptrons with two and three layers (MLP_2, MLP_3), and a random forest classifier (RF). We also consider using a two-layer perceptron (MLP_2) and a support vector machine (SVM) for the case of RSNA Bone Age (Gender). For this experiment and the rest of this section, we report results with KL Divergence Attack.

We observe several interesting trends for Census19 while varying model types for the victim and adversary (Table III). For instance, inference risk is slightly higher when the adversary uses models with high learning capacity like MLP_2, MLP_3 (n_{leaked} is 13.2), as opposed to ones with lower learning capacity like RF and LR (n_{leaked} is 10.4). Interestingly though, we also observe a sharp increase in inference risk when the victim uses models with low capacity, like linear regression and random forest instead of multi-layer perceptrons. For example, mean n_{leaked} is 5.9 when victim models have high learning capacity (MLP_2, MLP_3), and nearly triples to 16.4 when the victim models have low learning capacity (RF, LR). These trends hint at possible connections between distribution inference risk and model learning capacity.

B. Feature Extractors

When dealing with high-dimensionality datasets and a scarcity of data, it is common to use techniques such as transfer learning [26], [27] to boost model performance with reduced data and computational requirements. Using a pre-trained model for feature extraction should intuitively limit distribution-related privacy leakage, since there are fewer trainable parameters that can potentially contain revealing information. At the same time, fewer parameters also reduce the space of models, making it easier for an adversary to launch attacks. Even in a black-box setting, the adversary may be able to use the same feature extractor as the victim, either as a result of the adversary snooping and gaining information, or just by assuming the use of popular pre-trained models (like BERT [28]). This setting has been explored previously [1], [2], [5], but the effect of the adversary using the same or different extraction models from the victim model has not been previously explored.

Victim Model	Adversary Model FE+MLP_2	FE+SVM
Feature extractor, perceptron (FE+MLP_2)	3.0	0.7
Feature extractor, SVM (FE+SVM)	7.8	8.1
DenseNet (CNN)	0.4	<0.1

TABLE IV: n_{leaked} values for RSNA Bone Age (Sex), for different combinations of model types for victim and adversary

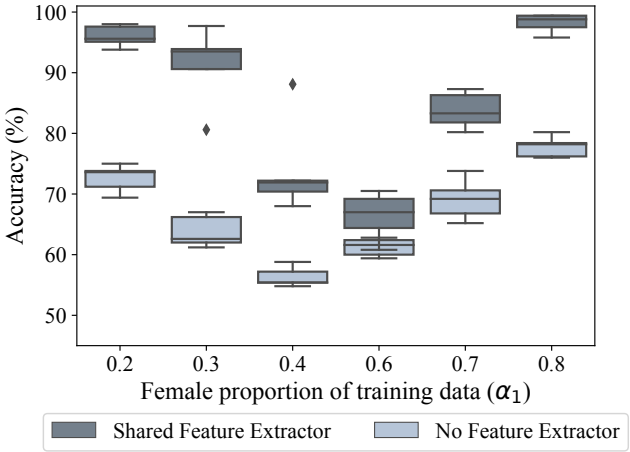

Fig. 3: Distinguishing accuracy for the KL Divergence Attack for RSNA Bone Age (Sex), when the adversary uses the same feature extractor as the victim, and when the victim does not use or share any pre-trained feature extractor. While there is an obvious drop in performance, inference risk still stays high.

For RSNA Bone Age (Sex), we consider two configurations: one where the victim and adversary use the same feature extractor, and another where the victim trains DenseNet [22] models from scratch (CNN). For the first setting, we explore an SVM (FE+SVM) and a two-layer perceptron (FE+MLP_2). There is a sharp drop in n_{leaked} from 3 to 0.4 when the victim and adversary no longer share feature extractors (Table IV). For the settings where they do, we observe leakage to be highest for similar model architectures, and note a sharp increase when the victim uses an SVM.

For CelebA (Sex), we explore a setting where the adversary utilizes a feature extractor to train its models, while the victim trains CNNs from scratch. This setup represents a resource-constrained adversary who uses pre-trained models to lower computational and data requirements. We observe a similar diminishing of inference risk when a shared feature extractor is not available to the adversary, consistent with the RSNA Bone Age results. Compared to the scenario where the adversary uses the same model architecture as the victim without any pre-trained feature extractors, n_{leaked} drops from 2.8 to <0.1.

C. Label-Only Access

Most black-box attacks in the literature related to distribution inference assume access to prediction confidence vectors.

Dataset/Task	Confidence Scores	Prediction Label
Census19 (Sex)	7.6	6.0
CelebA (Sex)	2.8	1.3
RSNA Bone Age (Age)	22.8	7.4

TABLE V: Effectiveness of label-only attacks. Each value is the observed n_{leaked} for the attack. The label-only setting leaks less information, but the attacks still are effective even when confidence scores are unavailable.

This is not an unreasonable assumption—many APIs return prediction scores, especially for top-k classes (for example Google Vision API and ClarifAI Prediction API return scaled confidence scores for the top 10 or 20 classes, respectively). It is unclear, however, what kind of performance drops to expect for distribution inference attacks in settings where the model's API only returns a label. The only previous work to explore distribution inference in the label-only was in the context of active adversaries with poisoning capabilities [3].

With some straightforward modifications, KL Divergence Attack can be launched with access to just label predictions, with negligible drops in inference leakage in most cases.

The attack requires prediction confidence scores to compute the KL-divergence values. However, these scores are absent in the label-only setting, and the labels effectively correspond to confidence values of 0 and 1. This makes the KL computations (Equation 1) invalid, since the log of 0 or $1/0$ is undefined. To tackle this, we replace $0/1$ labels with confidence scores ϵ and $1-\epsilon$ for some small value ϵ (set to 0.01 in our experiments).

We observe mixed trends across datasets and attacks. For instance, switching to the label-only setting has little impact in the case of Census19, while n_{leaked} values drop by more than half for CelebA (Table V). However, the drop in performance for CelebA is not uniform across all ratios. Inference risk is still quite high for many values of α (Figure 4). Similar trends hole for RSNA Bone Age, where distinguishing accuracy is $> 75\%$ for all ratios.

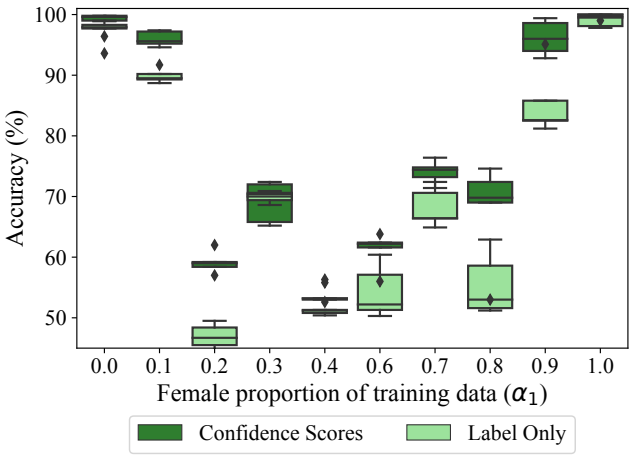

Fig. 4: Comparing the distinguishing accuracy for the KL Divergence Attack for CelebA (Sex), when the target model returns prediction confidence scores and when it returns only prediction labels. Performance drops most for certain ratios like 0.2 and 0.8, but remains high and roughly the same for more extreme ratios like 0.0, 0.1, and 1.0.

V. Defenses

Several defenses against distribution inference have been proposed, but most of them (except differential privacy [2], which has shortcomings as we discuss later) have not been actually evaluated. Like most defenses designed to limit privacy leakage, these defenses involve adding noise in some parts of the training process. This can include the data itself [1] or model parameters [2], [9]. We evaluate some of these noise-based defenses in Section V-A, and find that they seem unlikely to successfully mitigate distribution inference risks. Our exploration of inference risk with model generalization reveals interesting trends and a potential trade-off between learning and inference risk (Section V-B). Section V-C introduces and evaluates a simple defense based on data re-sampling, which can prevent distribution inference in settings where the model trainer knows which distributional property to hide.

Prior Work. Unlike membership inference where differentially private training can provide a guaranteed bound on inference risk, there are no defenses against distribution inference from trained models with theoretical guarantees. Chen and Ohrimenko [29] recently proposed a defense mechanism that builds upon formal notions of distributional privacy [30] to protect against property inference attacks on statistical queries. This is the first known theoretically-grounded defense against distribution inference, but it does not apply to protecting machine-learning models.

The only previous defense that has demonstrated meaningful protection against distribution inference attacks on machine learning models is NoSnoop [31], proposed for a collaborative learning setting. In their threat model, the adversary seeks to infer sensitive information about exact training batches and has access to intermediate model losses from clients. The defense works utilize a discriminator-generator setup, where gradient updates are used to minimize property leakage while preserving task-based utility. Although this defense is highly effective, it defends against a very narrow type of configuration, including properties limited to the presence/absence of sensitive data.

Recent work by Stock et al. [32] proposes a defense against distribution inference based on gradient updates from meta-classifiers. The victim, being aware of two distributions that an adversary may test for, trains multiple copies of its models with these training distributions. Then, it trains a meta-classifier and computes gradient updates for its own local models such that inference risk is minimized. This method requires the victim to train hundreds and thousands of models locally for the meta-classifier, leads to large drops in task performance, and does not generalize to settings where the victim and adversary use different kinds of meta-classifiers.

Other proposed defenses include removing sensitive attributes from features [6], using node-multiplicative transforms, or encoding arbitrary information into the models [2]. Since

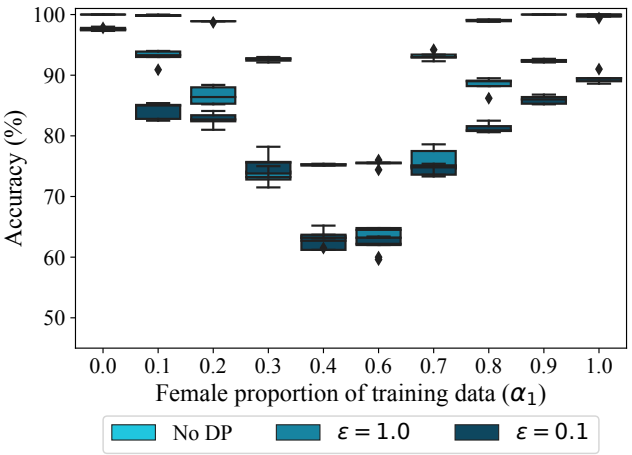

Fig. 5: Distinguishing accuracy for different for Census19 (Sex), using KL Divergence Attack. Attack accuracy drops with stronger DP guarantees (decreasing privacy budget ϵ).

the Affinity Graph Attack measures similarity between datapoints from one model at a time using a scale-invariant metric (cosine similarity), it would not be affected by any such node multiplicative transforms. Additionally, the Permutation-Invariant Network architecture can be modified to have some form of scale-invariance as well, which in turn can also bypass such multiplicative defenses. Further, these defenses seem unlikely to diminish black-box attacks, which our experiments have shown to be more effective than known white-box attacks, hence we do not evaluate them here.

A. Noise-Based Defenses

Several proposed defenses against distribution inference involve adding noise in various ways—differentially privacy training incorporates crafted noise in the training process and label poisoning adds noise to the training data. We also consider using adversarial training, which augments training with adversarial perturbations.

Differentially Private Training. Differential privacy (DP) is a formal privacy notion that provides theoretical guarantees that bound an adversary's ability to distinguish between neighboring input datasets from the output of a computation. Differential privacy can provide theoretical bounds limiting membership inference. Evaluations by Ateniese et al. [1] suggest differentially private training is not an effective defense against distribution inference attacks. However, their experiments used a setup with some overlap between the victim and adversary's data, so it is possible the observed lack of protection is related to the overlapping data available to the adversary.

We use DP-SGD [7] to train victim models with Reńyi Differential Privacy [33], with privacy loss budgets of $\epsilon = 1.0$ and $\epsilon = 0.1$, with $\delta = 1.3 \times 10^{-5}$. We evaluate this defense on Census19, since it is the only tabular dataset with non-trivial inference leakage. We observe a significant drop in n_{leaked} values, but inference risk stays high for ratios further away from $\alpha_0 = 0.5$ (Figure 5).

We hypothesize that this drop in effectiveness may not be because of the differential privacy noise itself, though, but could be because either the model does not learn the distribution well enough (and hence does not reveal it), or it produces arbitrary differences that cause a mismatch between the victim's models trained with using DP-SGD and the adversary's shadow models trained without privacy noise. Inspection of task accuracy for the differentially-privacy models suggests lower learning effectiveness as one potential factor (Table VIII). To test whether the decrease in prediction accuracy is mainly due to arbitrary differences in the models, we evaluate results for the setting where the adversary also trains its models using DP-SGD with the same privacy loss budget. Compared to an adversary that does not use DP, there is a clear increase in inference risk—n_{leaked} increases to 3.8 for $\epsilon = 1.0$, and 2.5 for $\epsilon = 0.1$ (compared to 7.6 without any DP). Assuming adversary's knowledge of the use of differentially-private training and the specific privacy loss budget is not a far-fetched assumption. Organizations that release differentially private models often document their exact levels of privacy budget [34], [35]. An adversary in such scenarios can thus train its models with the same privacy parameters.

Label Poisoning. Ganju et al. [2] proposed to mitigate distribution inference by adding noise to the training data via label poisoning. The underlying idea is to perturb the training data in a way that will alter the model parameters and make the adversary's task harder. Although changing data labels is prone to detrimentally harming the model's task performance, a model trainer may be able to find an acceptable trade-off between accuracy and inference risk. For a given noise ratio r, the defense comprises randomly flipping task labels for r fraction of the training data. We evaluate this defense for CelebA (Male) and RSNA Bone Age (Age) with a label noise ratio of 0.2, and Census19 (Gender) for label noise ratios 0.2 and 0.4. As expected, this defense harms task performance (Table VIII), reducing task accuracy: by $\sim 1-5\%$ for $r = 0.2$ for all three datasets, and $\sim 8\%$ for $r = 0.4$ on Census19. Average inference risk drops substantially for Census19, but remains is still quite high for ratios like $\alpha_1 < 0.2$ and $\alpha_1 > 0.8$, as shown in Figure 6. It may be possible to find a desirable tradeoff for a simple task like Census19, but this approach is not effective for more complex tasks. For instance, using a label noise ratio of 0.4 in CelebA completely destroys task performance, reducing the classifier to only slightly better than random guessing.

Adversarial Training. Adversarial training [8] involves using a training loss function that encourages the model to learn features that are robust to perturbations in the input, and produces models that are less prone to overfitting dataset-specific patterns [36]. This can be especially useful when the data includes properties that are not correlated with the task, and a model should not capture signals related to the irrelevant property. A model trained with adversarial robustness objective, using this reasoning, should be less susceptible to distribution inference. To test this hypothesis and explore

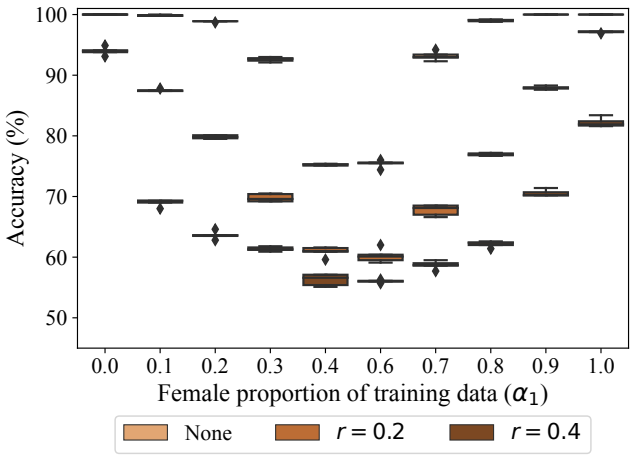

Fig. 6: Distinguishing accuracy for different for Census19 (Sex), for varying levels of label poisoning. Inference risk drops considerably with increasing levels of label poisoning, but is also followed with non-trivial drops in task accuracies.

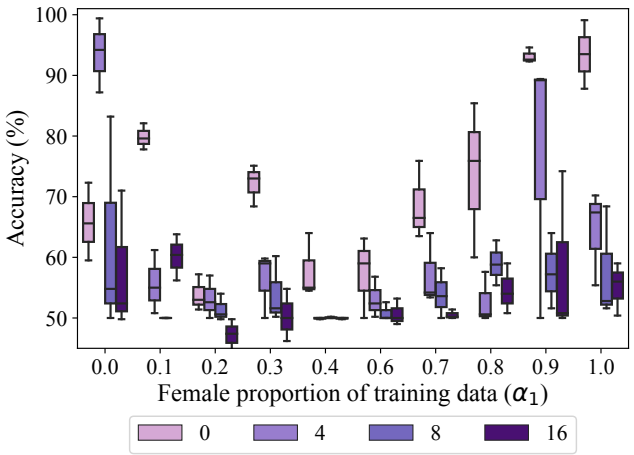

Fig. 7: Distinguishing accuracy for different using Affinity Graph Attack, for varying levels of adversarial robustness ϵ (/255) in L_∞ norm, for CelebA (Sex). Inference risk lowers with increasing robustness.

the impact of training for robustness. We train adversarially-robust models for varying L_∞ norms for the Gender and Age properties on CelebA, since the other datasets are either tabular or do not contain sufficient samples for adversarial training with acceptable performance. Figure 7 shows distinguishing accuracies for varying settings for the perturbation strength used in adversarial training. Training for adversarial robustness, as documented in the literature, leads to drops in task accuracy.

Interestingly, the KL Divergence Attack exhibits slightly worse performance than AGA (Table VIII). Nonetheless, inference risk drops significantly with adversarial training, and we observe a direct correlation between increasing perturbation strength (ϵ) and decreasing inference risk. The results in Figure 7 are using Affinity Graph Attack. Understanding this drop in performance suggests underlying distributional properties

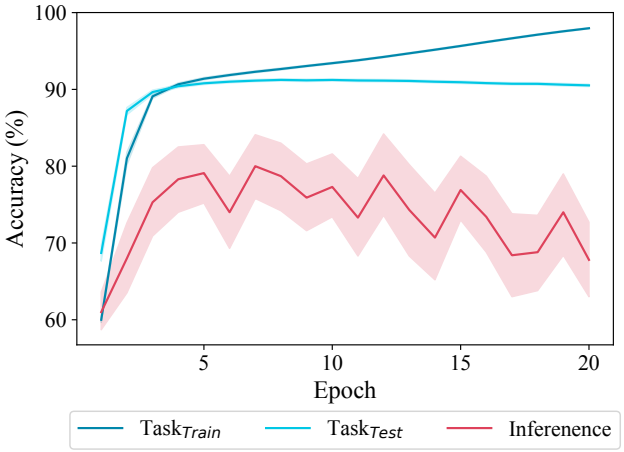

Fig. 8: Distinguishing accuracy of the KL Divergence Attack on CelebA (Sex), for varying number of training epochs for victim models. Shaded regions correspond to error bars. Distribution inference risk increases as the model trains, and then starts to decrease as the model starts to overfit.

may be captured differently under adversarial training, and is an interesting direction for future work.

Dataset/Task	**Adversarial Training** (ϵ)			
	0/255	4/255	8/255	16/255
CelebA (Sex)	0.7	0.4	<0.1	<0.1
CelebA (Age)	0.4	0.2	0.3	0.1

TABLE VI: Impact of adversarial training. Values are n_{leaked} for AGA on models trained with adversarial robustness, with varying norms ϵ (/255) of perturbation budget (L_∞ norm).

B. Generalization

The defenses discussed in Section V-A have one thing in common: they nearly always lead to non-trivial drops in task performance. This is not only unacceptable for most deployments, but raises the question of whether the defenses are doing anything useful or just reducing distribution inference by producing models that learn the underlying distribution less well. Concretely, we observe a positive correlation between model task accuracy and n_{leaked} values (Pearson's correlation coefficient > 0.55). A model with poor task performance possibly fails to learn useful signals from the training distribution, and is thus cannot leak properties it has not learned; while good performance means a model has learned the distribution well and is prone to more leakage.

To investigate these correlations, we inspect trends between inference risk and generalization across training epochs. For this experiment, we train the models longer than the other experiments (which were optimally selected for best generalization using validation data), allowing us to better study trends between overfitting and inference risk. We observe interesting trends in inference risk with model training. In most cases, inference risk is high after even one epoch of model training

(Figure 8). This is especially surprising because the model takes a few epochs to get good performance on the task itself, but shows that the model is learning and exposing aspects of the distribution even early in its training.

These trends clearly suggest that model under-training is not a feasible defense. Training beyond minimum generalization gap does lead to significantly reduced distribution inference risk. However, this region of the training corresponds to overfitting, which is known to be positively correlated with increased risk to membership inference [37]. Thus, a model trainer that is willing to overfit its models to avoid distribution inference adversaries would risk making the model more vulnerable to membership inference.

C. Re-Sampling Data

If the victim is aware of the property that an adversary might target, or only has a few known properties of the distribution that it wants to protect, the easiest mitigation is to just modify the training distribution (or the sampling mechanism) such that the property is no longer present for the training dataset. Knowing the particular property to hide is a plausible assumption that is often assumed in work on distribution inference defenses. For instance, Chen and Ohrimenko [29] propose a theoretically-grounded defense that builds upon the distributional privacy framework [38] and modifies feature values to provide privacy guarantees against distribution inference adversaries.

Zhou et al. [39] propose over-sampling to reduce inference risk, but do so by adding new samples to their training data. Although this defense eliminates distribution risk (at the cost of model performance), the availability of new data is not always possible. Model trainers typically use all available data. We explore two variations of re-sampling defenses: *over-sampling* and *under-sampling*. In both cases, the model trainer re-samples data from its available datasets such that the resulting dataset is indistinguishable from a dataset sampled from a different distribution. For over-sampling we experiment with two flavors: using simple replacement and over-sampling based on inserting augmented data. These defenses rely on the key assumption that the model trainer knows the property they want to hide, and that there are a only few such properties so re-sampling to hide the desired properties will not unduly hard the model's task performance. When this assumption holds, resampling defenses can virtually eliminate inference risk. We evaluate this defense on configurations with low (CelebA–Sex), medium (Census19–Sex), and high (RSNA Bone Age–Age) inference risk to measure the impact of this defense.

Under-Sampling. The model trainer can simply under-sample its data such that the resulting dataset has a ratio corresponding to some other distribution. For example, a model trainer, with a dataset containing 70% females who wants to hide the ratio of females in the dataset from an inference adversary can simply under-sample examples with the 'female' attribute such that its data is balanced between 'females' and 'not females'. An adversary seeking to distinguish between 70% and 50% could not distinguish between the cases where the victim's data actually was balanced to begin with, or was adjusted with this defense, even if given the training data set directly. Hence, this defense should prevent any disclosure about the pre-sampled distribution to even the strongest distribution inference adversary.

Analysis. Under-sampling lowers inference risk significantly, but does not completely eliminate it—n_{leaked} drops to ≤ 0.3 for CelebA and RSNA Bone Age, but stays high at 4.5 for Census19 (Table VIII). This is particularly surprising, since the ratios in both cases are same, and the only difference lies on the dataset sizes, which the adversary is unaware of.

Over-Sampling. A model trainer not willing to sacrifice available training data by under-sampling may prefer to over-sample. The most basic variant over-samples the data before training begins, duplicating training records, and then trains its models like usual. Although this defense leads to complete utilization of data, the presence of repeated data can be revealing and may reveal the property the adversary wants to hide to an adversary adversary aware of the defense. It could, for instance, lead to a change in group-wise accuracies, which an adversary can learn to identify and still succeed at distribution inference. Over-sampling further reduces inference risk for Census19, from n_{leaked} of 4.5 to 2.7 compared to under-sampling, but does not completely eliminate it. For the other two datasets, it is similarly effective in reducing leakage.

Augmentation Based Over-Sampling. The ideal scenario for the defense would comprise of injecting fresh labeled data to adjust the desired property, as was assumed by Zhou et al. [39]. However, labeled data is scarce and may be expensive to acquire, and using techniques like pseudo-labeling can still leak information. In such scenarios, the model trainer can use augmentation techniques to synthetically generate additional samples. This avoids repeating samples, and may have the added benefit of potentially increasing the model's robustness to augmentations. But, the use of augmented data in an imbalanced way may still reveal information to a distribution inference adversary. For this defense, we focus only on the CelebA dataset, since designing augmentation for tabular datasets is much harder, and augmentations for RSNA Bone Age are limited. Task accuracy remains comparable and inference risk drops significantly (slightly higher than other forms of sampling), but is not completely eliminated.

Impact on Fairness. This form of re-sampling is common in research related to improving fairness in machine learning [40], commonly known as "unbiasing". However, re-sampling data can impact different sub-groups and populations of the distributions unequally, creating issues related to fairness in model predictions [41]. To investigate such potential impacts, we measure the impact of under-sampling and over-sampling-based mitigation strategies on fairness. We compare the precision and recall for the group that is undersampled to equalize ratios for the undersampling defense to the group that is not undersampled. Re-sampling based defenses have negligible

Re-Sampling	α	Precision male	Precision female	Recall male	Recall female
Under-Sampling	< 0.5	$\downarrow 6\%$	$\uparrow 4\%$	$\uparrow 5\%$	$\sim$
	> 0.5	$\uparrow 4\%$	$\downarrow 2\%$	$\uparrow 1\%$	$\downarrow 9\%$
Over-Sampling	< 0.5	$\downarrow 7\%$	$\uparrow 2\%$	$\uparrow 6\%$	$\sim$
	> 0.5	$\uparrow 4\%$	$\downarrow 3\%$	$\uparrow 1\%$	$\downarrow 10\%$

TABLE VII: Relative change (%) in precision and recall metrics for various datasets, for RSNA Bone Age (Age) for under-sampling and over-sampling. We consider cases where data for males ($\alpha < 0.5$) or females ($\alpha > 0.5$) is under-sampled for equalization.

Defense	Task Accuracy (%)	n_{leaked} $\alpha_1 = 0.2$	n_{leaked} Mean
Census19 (Sex)			
No Defense	77.7 ± 0.5	6.7	7.6
DP ($\epsilon = 1.0$)	77.0 ± 0.5	2.0	2.0
DP ($\epsilon = 0.1$)	71.4 ± 0.7	1.2	1.2
Label Poisoning ($r = 0.2$)	76.9 ± 0.6	0.9	2.3
Label Poisoning ($r = 0.4$)	71.2 ± 1.2	0.2	0.4
Under-sampling*	77.5 ± 0.6	4.7	4.5
Over-sampling	77.3 ± 0.7	3.1	2.7
RSNA Bone Age (Age)			
No Defense	67.1 ± 2.0	20.1	22.8
Label Poisoning ($r = 0.2$)	63.8 ± 2.4	4.0	2.4
Under-sampling	65.9 ± 2.0	<0.1	<0.1
Over-sampling	65.6 ± 1.8	0.4	0.3
CelebA (Sex)			
No Defense	91.2 ± 0.8	0.1	2.8
Label Poisoning ($r = 0.2$)	89.3 ± 6.1	0.4	0.9
Adv. Training ($\epsilon = 4/255$)	90.1 ± 0.9	<0.1	0.4
Adv. Training ($\epsilon = 8/255$)	88.3 ± 1.2	<0.1	<0.1
Adv. Training ($\epsilon = 16/255$)	80.7 ± 2.4	<0.1	<0.1
Under-sampling	90.3 ± 1.1	0.5	0.3
Over-sampling	90.3 ± 0.9	0.3	0.1
Augmentation Based Over-Sampling	91.3 ± 0.8	0.1	0.4

TABLE VIII: Effectiveness of considered defenses. Each n_{leaked} value is the observed leakage with KL (except adversarial training, where we use AGA), for $\alpha_1 = 0.2$ and averaged over α values (Mean). Most noise-based defenses harm model task accuracies, and the only defenses that diminishes leakage without harming task accuracy are based on data re-sampling.

impact on fairness in the case of Census19 and CelebA, but result in disparate impacts of both under/over-sampling on the two groups. for RSNA Bone Age (Table VII). For instance, under-sampling from a ratio $\alpha > 0.5$ lowers both precision and recall for females, while lowering both metrics for males.

VI. Limitations and Conclusions

Distribution inference attacks are known to reveal sensitive properties about underlying training distributions, but their effectiveness has been established only in very controlled settings so far. Our work advances understanding of this risk in more realistic settings, but is still far from understanding all of the issues that might impact a real deployment and attack.

Our proposed black-box attacks are highly efficient and maintain their effectiveness even when access to exact prediction probabilities is unavailable. Our experiments find that differences in model architectures harm inference accuracies, but the impact is not very severe as long as models with similar complexity are used. But the lack of common feature extractors, a common setting in many evaluations in the literature, virtually eliminates inference risk.

Although the white-box AGA outperforms the current state-of-the-art white-box Permutation Invariant Networks, it is often outperformed by the black-box KL attack. This suggests the methods we tried of combining the two attacks have room for improvement, and that there is still information available to a white-box adversary unexploited by any known attack.

Like nearly all inference privacy work, we assume an adversary with access to a dataset that matches the underlying distribution (in this case, before the transformation to the actual training distribution as modeled by $\mathcal{G}_0$ and $\mathcal{G}_1$). This is a strong assumption, which may be realistic in some cases but is often unlikely. All our attacks (and nearly all previous ones) require representative data for training models locally. Exploring adversaries with limited data access to these distributions and how it impacts inference risk is left as part of future work.

Configurations beyond single-party learning have seen growing interesting lately. Active adversaries with data poisoning capabilities have been demonstrated to be highly effective in both single-party [3] and multi-party [42] settings. Our proposed attacks are highly potent for many configurations, but the exact impact of different training setups like federated learning remains largely unexplored.

The general approach to achieve security and privacy for machine-learning models is to add noise, but our evaluations suggest this approach is not a principled or effective defense against distribution inference. The main reductions in inference accuracy that result from these defenses seem to be due to the way they disrupt the model from learning the distribution well, so observed reductions in inference risk are related to drops in task performance. Our experiments with different model architectures and differentially private training support this—inference risk increases significantly when the victim and adversary use the same learning algorithms or model architectures. The only reliably effective defense from our experiments is to re-sample data, which depends on the assumption that the model training is aware of the adversary's inference goals (or at least of the properties that should be protected). These re-sampling defenses too are not perfect, however, as they seem to negatively impact fairness of groups related to the property attribute.

There is a need for more theoretical connections between distribution inference risk and general useful notions of machine learning, like model complexity and fairness. Our work suggests such connections do exist, and we hope they will be better understood as both empirical and theoretical understanding of inference privacy advances.

*We noticed problems with re-sampling experiments for Census19 in preparing the paper, but did not have time to correct and re-run them before the deadline.

REFERENCES

[1] 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.

[2] 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.

[3] S. Mahloujifar, E. Ghosh, and M. Chase, "Property Inference from Poisoning," in *IEEE Symposium on Security and Privacy*, 2022.

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

[5] A. Suri and D. Evans, "Formalizing and Estimating Distribution Inference Risks," in *Privacy Enhancing Technologies Symposium*, 2022.

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

[7] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations*, 2018.

[9] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," in *IEEE Symposium on Security and Privacy*, 2019.

[10] 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.

[11] A. Suri, P. Kanani, V. J. Marathe, and D. W. Peterson, "Subject Membership Inference Attacks in Federated Learning," *arXiv preprint arXiv:2206.03317*, 2022.

[12] J. Hartley and S. A. Tsaftaris, "Measuring Unintended Memorisation of Unique Private Features in Neural Networks," *arXiv preprint arXiv:2202.08099*, 2022.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[14] H. Sak, A. W. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," in *INTERSPEECH*, 2014.

[15] J. Su, "Census19," https://github.com/JerrySu11/CensusData, 2022.

[16] 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*, 2000.

[17] B. Jayaraman, "Texas-100X," https://github.com/bargavj/Texas-100X, 2022.

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

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[20] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning Face Representation from Scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[21] 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.

[22] 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.

[23] 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*, 2020.

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

[25] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems*, 2015.

[26] Y. Xie and D. Richmond, "Pre-training on Grayscale ImageNet Improves Medical Image Classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[27] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models are Zero-Shot Learners," in *International Conference on Learning Representations*, 2021.

[28] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[29] M. Chen and O. Ohrimenko, "Protecting Global Properties of Datasets with Distribution Privacy Mechanisms," *arXiv preprint arXiv:2207.08367*, 2022.

[30] W. Zhang, O. Ohrimenko, and R. Cummings, "Attribute Privacy: Framework and Mechanisms," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022, pp. 757–766.

[31] X. Ma, B. Li, Q. Jiang, Y. Chen, S. Gao, and J. Ma, "NOSnoop: an Effective Collaborative Meta-Learning Scheme against Property Inference Attack," *IEEE Internet of Things Journal*, 2021.

[32] J. Stock, J. Wettlaufer, D. Demmler, and H. Federrath, "Property Unlearning: A Defense Strategy Against Property Inference Attacks," *arXiv preprint arXiv:2205.08821*, 2022.

[33] I. Mironov, "Rényi Differential Privacy," in *IEEE Computer Security Foundations Symposium (CSF)*. IEEE, 2017.

[34] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, and D. Davidson, "Learning new words," *Granted US Patents*, vol. 9594741, 2017.

[35] J. M. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, R. Johns, D. Kifer, P. Leclerc, A. Machanavajjhala *et al.*, "The 2020 Census Disclosure Avoidance System TopDown Algorithm," *arXiv preprint arXiv:2204.08986*, 2022.

[36] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial Examples are not Bugs, they are Features," *Advances in neural information processing systems*, vol. 32, 2019.

[37] 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.

[38] Y. Kawamoto and T. Murakami, "Local Obfuscation Mechanisms for Hiding Probability Distributions," in *European Symposium on Research in Computer Security*. Springer, 2019, pp. 128–148.

[39] J. Zhou, Y. Chen, C. Shen, and Y. Zhang, "Property Inference Attacks Against GANs," *NDSS*, 2022.

[40] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[41] E. Krasanakis, E. Spyromitros-Xioufis, S. Papadopoulos, and Y. Kompatsiaris, "Adaptive Sensitive Reweighting to Mitigate Bias in Fairness-aware Classification," in *Proceedings of the 2018 world wide web conference*, 2018.

[42] Z. Wang, Y. Huang, M. Song, L. Wu, F. Xue, and K. Ren, "Poisoning-Assisted Property Inference Attack Against Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.